

Mixture Models of Endhost Network Traffic

John Mark Agosta
Intel Research

Daniel Ting
U.C. Berkeley

Jaideep Chandrashekar
Intel Research

Mark Crovella
Boston University

Nina Taft
Intel Research

ABSTRACT

In recent years there has been much interest in modeling internet traffic that comes from inside large networks, such as at routers or gateways. In this work we focus on modeling a little studied type of traffic—namely the network traffic generated from endhosts. We study traffic data collected from hundreds of enterprise laptop users. We introduce a parsimonious parametric model of the marginal distribution for connection arrivals. We employ mixture models based on a convex combination of component distributions with both heavy and light-tails. These models can be fitted with high accuracy using maximum likelihood techniques.

Our methodology assumes that the underlying user data can be fitted to one of many modeling options, and we apply Bayesian model selection criteria as a rigorous way to choose the preferred combination of components. Our experiments show that a simple Pareto-exponential mixture model is preferred for a wide range of users, over both simpler and more complex alternatives. This model has the desirable property of modeling the entire distribution, effectively segmenting the traffic into the heavy-tailed as well as the non-heavy-tailed components. Scaling the time-window used to bin the data varies the relative contributions of the components strongly, but affects the component parameters less so. We illustrate that this technique has the flexibility to capture the wide diversity of user behaviors.

1. INTRODUCTION

In the last decade or so, there has been a tremendous amount of research done in the area of Internet traffic modeling (e.g., [7–9, 19, 21, 24, 31, 33, 37, 38]). Traffic models are helpful in solving a wide range of problems, including traffic engineering, service provisioning, routing, and network performance evaluation. To date, however, the vast majority of traffic modeling research has focused on traffic seen inside a network: at routers, gateways, or servers. Relatively little work has been done to model traffic as seen at endhosts, e.g., at laptops or desktops.

The paucity of endhost traffic models is limiting, because many problems can benefit from an understanding of the nature of endhost traffic. For example, a natural formulation for synthetic traffic

generation is as the superposition of traffic from a collection of user models. In that approach, traffic volumes can be scaled by varying the number of synthetic users, and traffic mixes may be varied by varying the behavior of individually modeled users.

The most likely reason that endhost traffic models are so scarce is that it is difficult to obtain the raw measurements needed, since those measurements require the express consent of each user in a sufficiently large set. Furthermore, such measurements essentially require installing a collection tool directly on each user’s machine — a tool whose management requires considerable goodwill from the affected users.

The value of endhost models combined with the difficulty of endhost instrumentation have motivated some efforts that have tried to infer endhost traffic properties from an observation point inside the network [15, 17]. While such approaches have shed some light, they are fundamentally limited — for example, when users are mobile. What is needed for a comprehensive view of endhost traffic is a measurement tool that moves with the user and continues to observe network traffic as the user switches between different networks and different environments (e.g., work and home).

In this work we deploy such a tool and analyze its outputs to develop models for end user traffic. We study a population of 270 enterprise users over a period of five weeks (§3). Our tool collects all packet headers entering and exiting the machine, on all networking interfaces. To accomplish this, we solicited enterprise employees to sign up on a voluntary basis for the trace collection. Participants explicitly gave consent for data collection; each user downloaded and installed the data collection software on their personal machines.

The most salient property of the resulting traffic measurements is the presence of *heavy tails* (data that seems well modeled by a power law distribution). Among other results, we show for the first time that the counts of flow arrivals binned over a range of intervals show heavy-tailed behavior. We also find heavy tails in a number of other traffic properties, some of which echo and deepen results from prior work.

While many studies have looked at heavy tails in network traffic, there are a number of limitations in previous studies that we overcome in this work. First, many previous studies have developed models or parametrizations that only describe the tail component of some metric. Our goal was to build models for traffic properties that describe the entire distribution, not just the tail. This is important, for example, in generating synthetic traffic for performance evaluation. When studying a load balancing or resource allocation scheme, it is often of interest to be able to compute the entire distribution of performance measures (such as delay), in addition to studying low probability events (such as loss).

The second limitation we overcome concerns *goodness-of-fit test-*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

ing versus model selection. Fitting power law models to data is common, though the process is notoriously tricky [6, 7, 25, 28, 29]. Frequently the experimenter will assume a power law form, confirm it by visually observing the data on a log-log plot, and then extract the slope parameter by performing a least-squares regression on the logarithm of the histogram. Given parameter estimations, the next step is to pick a power law distribution and forms a hypothesis that the data could have been plausibly generated from this distribution. To test the hypothesis, goodness-of-fit methods are often used. The limitation of this approach is that goodness-of-fit tests, and their associated P -values, are meant to rule out hypotheses (i.e. to reject the hypothesis). This is certainly useful for guiding data investigation, but they do not actually tell us when the model is right. The best one can do with this method, in a statistical sense, is to say that a given model has not been ruled out by the data. In fact, there will be cases when more than one model passes a goodness-of-fit test, or none of the proposed models do, resulting in an indeterminate result. What is needed to confirm the proper choice of model is not model fitting but rather model selection, a different problem with a different statistical basis.

In [6], the authors highlight the lack of care pervasive in the literature on power laws, and apply a rigorous approach to applying goodness of fit methods. In the process they review numerous power law claims that have been made in the literature, and find that claims of power law tails among well-known supposedly-“power law” datasets are not supported by the data. The clear conclusion from that work is the indeterminacy of goodness-of-fit methods for applying power law models to data.

To address this limitation, we approach the modeling problem as model selection rather than as goodness-of-fit testing. The goal of model selection is to compare models from a given set, as opposed to merely estimating the best fit for a single model. Thus we do not presuppose a particular distribution model; instead we start with a family of *nested* models (i.e. a family of distributions where one is a subset of another) and our methodology selects the best model. To select among the candidate models, we use *Bayes Factors* to make pairwise comparisons of models on the same set of data, and to make specific statements about which model is better [23]. A requirement for using Bayes Factors is to consider models that apply to all the data (not just the distributional tail). For the large sample sizes that we consider, the log Bayes Factor can be well approximated by the difference of two models’ *Bayesian Information Criteria* (BIC). Pairwise examination of models’ BIC values allows the comparison of increasingly complex models (in terms of the number of parameters), while factoring out the improvement in fit that is merely due to increased degrees of freedom. When two models are compared and little difference is observed, then the less complex of the two can be selected. The BIC model selection criterion penalizes a model based on the number of its parameters, and thus balances the cost of an increased number of parameters against the improvement in goodness of fit.

This work makes a number of discrete contributions. Our first contribution is in modeling endhost traffic using mixture models (§4.1), and in doing so using a rigorous model selection approach. We have elected to study mixture models because they are flexible enough to model complete distributions for our data (rather than just the tail). A mixture model is a convex combination of “component” distributions, where the parameters of the component distributions as well as the mixture variable (itself a discrete probability distribution) are estimated from data. We use both conventional component distributions (exponential) as well as power law (Pareto) components. The nested family of distributions we consider are: Pareto-only models, denoted P; a mixture of one expo-

ponential and one Pareto, denoted EP; and a mixture of 2 exponentials and 1 Pareto, denoted EEP.

Our second contribution is the application of Bayes Factors for model selection (§4.3). Although this method is well known in the statistics community, it has seen little applicability in the Internet measurement community. We use Maximum Likelihood methods (§4.2) for parameter estimation and validate (§5) the accuracy of our parameter estimation technique on synthetic data created from mixture models. Our success with this method in modeling endhost traffic suggests that it might be fruitful to explore using this modeling technique to other heavy-tailed datasets of network measurements.

Our third contribution lies in the results of extensive application of this method on our endhost traffic data (§6). We find that for the metrics we study (flow arrivals and idle period lengths), the vast majority of users are well modeled by the EP distribution; a much smaller number are better modeled by P and EEP models. Here the flexibility of our approach is a strength, because our method does not insist that all users need to be described by the same model.

Our final contribution lies in examining the results of our modeling. We expose and highlight strong invariants across users (§6), but also illustrate the nature of the diversity among different users. For example, we demonstrate that tail properties and mixing fractions differ dramatically across our set of users. Finally in §7 we discuss application of our results to synthetic traffic generation, and we provide an initial exploration of the generative processes at the application level that may influence the nature of the resulting mixture models.

2. RELATED WORK

Heavy tailed statistics have been documented in numerous phenomena in network traffic; in the popularity of web pages [5], in traffic demands [13]; in network topology [25], in TCP inter-arrival times [12], in wireless LAN traffic [26], and many others. As mentioned earlier, most of this work analyzes traffic collected from inside the network at locations where anywhere from hundreds to millions of users’ traffic is aggregated. To the best of our knowledge, our work is the first to study connection traffic generated directly on user laptops.

The seminal work by Leland et al. [24] studied LAN traffic and convincingly demonstrated that actual network traffic is self-similar or long-range dependent in nature (i.e., bursty over a wide range of time scales). Our work differs in two ways. First, that study’s Ethernet LAN data captured the aggregated traffic of many users, whereas we focus on models for individual user traffic. Second, we observe the power law nature of traffic in the first-order statistics of traffic rates, rather than in the second-order autocorrelation properties. Both approaches result in estimating a power-law parameter, but the meaning of the parameters should not be confused.

Modeling and characterization studies are often followed by studies that attempt to understand the underlying causes for the observed behavior [8, 19, 38, 40]. Our study is mainly concerned with finding models for endhost traffic, although we do briefly explore some questions of generative processes. In future work we hope to more deeply investigate the causes lying behind EP mixture models.

Work closer to our study is reported in [2, 10]. Those studies captured HTTP requests through instrumentation in web browsers or proxies, and so are similar to ours in focusing on a traffic seen at a fixed set of endhosts. Results from those studies were used in developing tools for generating representative user-level HTTP traffic [3]. However, those studies did not look at the total set of an individual endhost’s pattern of connections over time. This crucial

difference makes our results more useful for general traffic modeling. In particular, those studies only looked at Web traffic (a subset of all traffic on the endhost) meaning that the resulting traffic generation tools were limited to reproducing only Web browsing behavior. In fact, as we shall see, the aggregate traffic on endhosts is influenced strongly by applications other than the Web. An important aspect of this distinction is that our data also includes network traffic that is machine generated (i.e. not user generated). Machine generated traffic comes from background enterprise applications, chatty protocols, and the many auto-update checking mechanisms (e.g., for software and firewall rule updates) that are typically installed on corporate laptops.

Another end user study looks at data from Neti@Home users, and models think time as well as bytes sent and received for TCP and UDP connections [34]. They do not model traffic at the granularity of connection arrivals. In [4], the authors report on the diversity in distributional tails of user behaviors. This diversity is captured by a simple metric, the 99th percentile of various distributions on user protocol traffic. The models proposed in our paper capture user tail diversity with richer measures, such as the slope parameter α of the Pareto distribution from the EP model.

The idea of using mixture models for Internet traffic has been proposed in other contexts before [14]. That work proposes using hyperexponential models to approximate heavy-tailed distributions. Thus it is not about explicitly modeling data collected from the Internet, but more about fundamental methods for approximations of heavy-tailed distributions. The advantage of their work is that their effort provides analytically tractable representations that can be used subsequently for queueing theory models. However, the disadvantage (as the authors acknowledge) is that their mixture models have a large number of parameters. In our work, we obtain parsimonious models with a small number of parameters. All of our models range from having 1 to at most 5 parameters; most users are well modeled using only 3 parameters. Further, in contrast to the fitting-oriented approach [14], our work not assume the presence of a heavy-tailed component ahead of time. It is entirely possible for our mixture model to assign a negligible Pareto component to a dataset.

One of the driving motivations for developing traffic models is to be able to build traffic generators that can be used in simulation and emulation tested beds for evaluating protocols, traffic engineering optimizations, and so on. A good list of available tools can be found at [1]. Most of these tools generate aggregated traffic; in contrast, models such as ours could be used to generate single user traffic as well. For example, SWING incorporates some aspects of user behavior [36]. SWING combines user, application and network behaviors to generate accurate request-response streams inside a network; it successfully reproduces the types of burstiness seen in the Internet today. The SWING user model is primarily focused on capturing user idle and active times, and does not capture the traffic metrics that we do, namely connection arrivals. We anticipate that our modeling results could be incorporated into SWING to enrich the component of that system that describes user behavior.

Recently there is increased interest in modeling and describing the behavior of enterprise end users [15, 16]. IT management is driving this trend, as it faces an increasingly heterogeneous computing environment. Autonomic computing is heading towards self-diagnosis for fault identification, and endhost profiles are being explored for security purposes [4] and resource management. For example, in [22] the authors design mechanisms to allow hosts to participate in network management, traffic engineering and other operational decisions by explicitly controlling host traffic. To better calibrate such applications, a deep understanding of end user

traffic is needed. Our study is one step in this direction.

3. DATASET DESCRIPTION

The dataset used in this paper consists of traces collected at 270 enterprise end-hosts (90% laptops), spanning a period of approximately 5 weeks. Because of the way our enterprise manages employee machines, each end-host corresponds to one and only one user. All of the participants ran a corporate standard build of Windows XP on their hosts that includes a number applications required by the enterprise IT organization for security, maintenance and monitoring.

Packet-level traces were collected *on* the end-hosts, rather than at a network tap. This provides visibility into *all* the traffic, even as end-hosts move in and out of the network, change interfaces, and/or IP addresses. Our software collection tool used WinDump, a Windows version of tcpdump, to log packet headers. Concurrently, we ran a homegrown application that sampled user-activity variables every second such as the number of key strokes, number of mouse clicks, and CPU load. Both WinDump and our activity monitoring code were wrapped in an application that tracked changes in the active IP address and network interface, and that opportunistically uploaded trace files to a central server. Importantly, the logging was turned off during uploads. The trace collection effort yielded approximately 400 Gb of packet header traces.

These packet level traces were converted into flows (based upon the standard 5-tuples) using BRO [32]. The starting time of each flow generates a point process in continuous time that we bin over non-overlapping constant duration time-windows to create a time series for each user. Each user trace was binned for 8 different window sizes, starting at 4 seconds, and increasing in multiples of 2, up to 512 seconds. Each bin contains a count of the new flow arrivals. The *flow count* events within each time-window or *bin* are the random variables modeled in this work. In our datasets the median sample size was 9771 intervals, and the maximum was 264000. There can be two causes for the appearance of no network activity in some bins of a trace. One is that the machine is off (or in standby). These periods can easily be confirmed in our traces because we can check that the CPU load was zero, and that there were no mouse or keyboard clicks. We removed these times from our traces because we are interested in analyzing network traffic when the machine is powered on. The second cause for inactivity occurs when the machine is active but there are simply no new flows generated. We identified these periods using the user-activity trace associated with the packet trace to infer those intervals when the end-host was connected to the network, but did not generate new flows. For these time intervals, the counts in a bin are zero.

We mainly focus on modeling the flow events when the counts are nonzero. We do this in order to characterize the flow traffic when the network is active. However, for the purposes of synthetic traffic generation it is important to understand the network-idle times. We thus processed our time series to gather the length of all the network-active and network-idle periods per user. In section 7, we illustrate that our methodology can also be applied to parametrize an ON-OFF model for network-activity and network-idle.

4. METHODOLOGY

4.1 Mixture Models with Heavy Tails

A *mixture model* is a probability model composed of a convex combinations of probability densities. Such models are familiar in the Statistics literature, [11] [18] and have gained recent popularity

due to their similarity to some artificial neural network models and their possible Bayesian interpretation [20]. A mixture model can be thought of as a hierarchical model where the mixing weights determine the probability of each of the component models.

We first introduce some notation. For component densities, $f_i(x)$, and mixture fractions m_i , the finite mixture model, of k components, with parameters \mathbf{m}, θ is the convex combination given by:

$$f(x | \mathbf{m}, \theta) = \sum_{i=1}^k m_i f_i(x | \theta_i), \quad (1)$$

$$\text{s.t. } \sum_{i=1}^k m_i = 1, m_i > 0.$$

where the θ are the component parameters, and $\mathbf{m} = m_1 \dots m_k$. The *degrees of freedom* of the model is then just the count of parameters, e.g. for k components, each with a single parameter, the full model will have $k + (k - 1)$ parameters, the -1 because of the normalization constraint on \mathbf{m} .

We propose to consider the following nested family of distributions: a Pareto only model labeled (P), a mixture of one exponential and one Pareto (EP), and a mixture of two exponentials and one Pareto (EEP). The “pure power-law” model we fit is

$$f(x | \alpha) = Cx^{-\alpha}, \quad (\text{P})$$

Because flow counts take on positive integer values, we use the discrete version of the Pareto density in our models. The value of the normalizing constant, C for the discrete Pareto (referred to also as the *Zeta*) is

$$C = \frac{1}{\zeta(\alpha, x_{min})}, \quad (2)$$

where

$$\zeta(\alpha, x_{min}) = \sum_{n=0}^{\infty} (n + x_{min})^{-\alpha}. \quad (3)$$

Our P model serves as the “null hypothesis” in our model selection approach. Our EP model is defined as

$$f(x | \mathbf{m}, \lambda_1, \alpha) = m_1 \lambda_1 e^{-\lambda_1 x} + (1 - m_1) C x^{-\alpha}. \quad (\text{EP})$$

The motivation to using such a model is so the tail behavior can be captured by the Pareto component, thereby allowing the properties of the central part of the density to be revealed and captured by an alternate distribution. The mixture variable adds another degree of freedom, revealing the relative contribution of the components. The two exponential mixture density model is thus:

$$f(x | \mathbf{m}, \lambda_1, \lambda_2, \alpha) = m_1 \lambda_1 e^{-\lambda_1 x} + m_2 \lambda_2 e^{-\lambda_2 x} + (1 - m_1 - m_2) C x^{-\alpha}. \quad (\text{EEP})$$

The intent behind using a family of models is to allow for the diversity of each user’s machine to be captured. These models allow for data that is either purely heavy-tailed or is a mixture. Note that they can also capture data with little to no heavy-tail. In such cases, an EP model could be selected with a insubstantial weight for the Pareto component. In general, these are very parsimonious models; the EP model has 3 parameters, and the EEP has only 5. We’ve set the Pareto parameter x_{min} (indicating where the tail starts) to one because is the smallest value our flow count data can take on when the network is active, and we need the model to span the range of

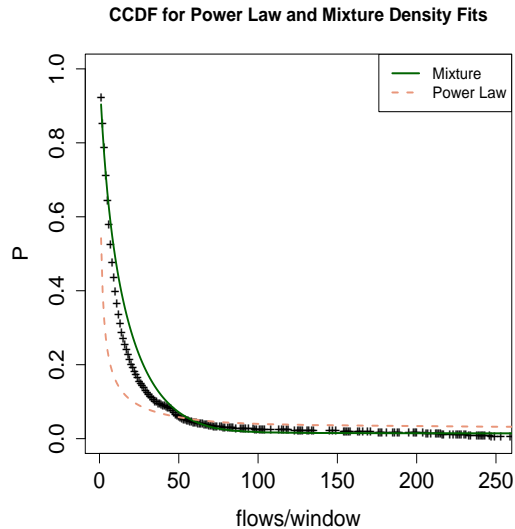


Figure 1: Flow count data for a single endhost machine. Visualization of EP and P model fits.

data values. We may still recover a threshold for where the tail starts by examining where the pareto mixture component becomes dominant.

We were motivated to consider these models not only because they capture a broad range of traffic behavior and because they are parsimonious, but also because our visual exploration of the data indicated the potential was good. An example of data from a single user machine is plotted in Figure 1. We fitted both an EP model (solid colored line) and a Pareto distribution (dashed line). We can see that the exponential component of the mixture model shows a close match for the dense part of the distribution (upper left) whereas the pareto exhibits a poor fit for this portion of the data. For the tail portion, both models do reasonably, although the EP model is a bit better.

4.2 Estimating Model Parameters

The model parameters are estimated using maximum likelihood. The maximum likelihood estimate (MLE) has numerous attractive qualities. If the model contains the true data generating distribution, and is differentiable in quadratic mean (DQM) [35], the MLE converges to the true parameters at a rate $O(1/\sqrt{n})$. Pareto distributions and mixtures of DQM models satisfy differentiability in quadratic mean. Even if the model does not contain the true data generating distribution, the MLE converges to the best approximation to the true distribution within the model’s constraints at a rate $O(1/\sqrt{n})$. The MLE is also asymptotically efficient, so no other estimator can obtain a better asymptotic variance than the MLE.

We used an interior point method [39] to enforce the constraints on the model parameters. Interior point methods are iterative optimization methods that enforce constraints by adding a weighted concave barrier function that steeply decreases to $-\infty$ at the boundary of the constraint set. This concave barrier prevents Hessian based maximization methods from stepping outside the constraint set. The weight on the barrier is decreased while using the previous solution for initialization, and a new solution is computed. The weight continues to be reduced until the barrier becomes negligible. A typical choice of barrier function is log. Thus, for the EP model with likelihood function l , mixing weights m_1, m_2 , pareto param-

eter α , and exponential parameter λ , the constrained optimization problem

$$\max_{\substack{m_1+m_2=1 \\ \alpha>1, \lambda>0}} l(m_1, m_2, \alpha, \lambda; x)$$

may be solved by the sequence of unconstrained problems

$$\begin{aligned} \max_{m_1, \alpha, \lambda} \quad & l(m_1, 1 - m_1, \alpha, \lambda; x) + c_1^{(t)} \log(\alpha - 1) \\ & + c_2^{(t)} \log(\lambda) + c_3^{(t)} \log(1 - m_1) + c_3^{(t)} \log(m_1) \end{aligned}$$

where m_2 has been replaced by $1 - m_1$ and the weights on the barrier $c_i^{(t)} \rightarrow 0^+$ as $t \rightarrow \infty$. By convention, we take $\log(x) = -\infty$ if $x \leq 0$. These unconstrained problems can be solved using the `optim()` function in the statistical programming language R. This function implements a Quasi-Newton method for optimization. To exclude obviously bad solutions, we also added constraints that the Pareto and the exponential parameters were not too large with $\alpha < 4$ and $\lambda < 3.5$.

Since the mixture model typically contains local optima, we performed the optimization multiple times with random initializations to find the global maximum. We also used small initial values of $c_i = 0.01$ for the regularization parameters and reduced them to $c_i = 10^{-8}$ in 3 steps to prevent the initial unconstrained problem and regularization path from unduly influencing the search for the global maximum.

4.3 Model Selection

We now describe the method we use to select the best model from our set of three. Each of the 3 models is fitted to the data from a single endhost and given a score indicating how well the model explains the data. From the Bayesian perspective, the better model is the one that is more probable for the set of data observed. Model selection is thus a question of comparing which model is more probable. We adopt the method for comparing two models in [23], and based upon the numerical value of the comparative metric, we decide which is better. If they are indistinguishable, then we select the model with fewer parameters.

We first explain our use of Bayes Factors – a method related to likelihood ratios, but with a Bayesian flavor – as a rigorous method for model selection. This section follows closely the presentation in Kass and Raftery’s recent paper [23]. We view model selection as computing the model \mathcal{M} that is most probable given the data, \mathcal{D} . This requires computing the posterior of the model evaluated at parameter values θ : $P(\mathcal{M} | \mathcal{D}, \theta)$. For purposes of comparison we want to find the model, within the range of allowable parameter values, that maximizes this probability - or equivalently, to find the maximizer of the log posterior:

$$\hat{\theta} = \operatorname{argmax}_{a < \theta < b} \log P(\mathcal{M} | \mathcal{D}, \theta) \quad (4)$$

Posteriors are computed with Bayes rule. In practice, numerical issues arise since the likelihood term becomes infinitesimally small as the number of data points becomes large, and we consider instead the posterior odds —the ratios of posteriors—between two models, as our selection criterion. To compare the model \mathcal{M}_P to the proposed model \mathcal{M}_{EP} , the posterior odds will be

$$\frac{P(\mathcal{M}_{EP} | \mathcal{D})}{P(\mathcal{M}_P | \mathcal{D})} = \frac{P(\mathcal{D} | \mathcal{M}_{EP}) P(\mathcal{M}_{EP})}{P(\mathcal{D} | \mathcal{M}_P) P(\mathcal{M}_P)} \quad (5)$$

The Bayes factor, BF , in this equation is defined as the ratio of

marginal likelihoods:

$$BF_{EP,P} = \frac{P(\mathcal{D} | \mathcal{M}_{EP})}{P(\mathcal{D} | \mathcal{M}_P)} \quad (6)$$

An unprejudiced rule implies equal model priors, in which case the Bayes Factor and the posterior odds-ratio are equal. Note that this criterion is similar to a maximum likelihood ratio, but rather than taking the probability at the maximum, the integral of the probability over the range of θ is needed. Doing so results in a correction on the degrees of freedom in the model which guards against overfitting. Adding more parameters to a model and thus increasing its degrees of freedom can only increase the likelihood at the maximum but does not necessarily improve marginal likelihood.

This integral requires a prior over the θ . Experiments on our large sample data showed that likelihoods are strongly peaked around their maximum at $\hat{\theta}$, not surprisingly. Thus numerical integration works poorly due to limited numerical precision. Both the questions about integration and the prior for strongly peaked likelihoods may be resolved by using an approximation to the likelihood integral using the Laplace approximation, which can be thought of as a Taylor expansion of the log likelihood computed at the MLE.¹ The Laplace approximation can be further approximated by the Bayes Information Criterion (BIC). BIC is often presented as a correction to maximum log likelihood to account for the degrees of freedom of a model. The BIC is defined as

$$\text{BIC} = \log P(\mathcal{D} | \mathcal{M}, \hat{\theta}) - \log(N) \cdot d/2 \quad (7)$$

where N is the sample size and d is the numbers of parameters in the model. Note that since the likelihood term $\log P(\mathcal{D} | \mathcal{M}, \hat{\theta})$ scales as $\mathbf{O}(N)$ with sample size, so does the BIC. In our experimental work we computed both Laplace approximations and BIC corrections and found to our satisfaction that they agreed to within a fraction of a percent on the data used.

With the BIC approximation, the log Bayes Factor becomes

$$\log BF_{EP,P} = \text{BIC}_{EP} - \text{BIC}_P \quad (8)$$

When computing Bayes factors for different bin sizes, the number of bins comprising the sample size varies. To remove this effect, we divide by the sample size N . Of course, in interpreting the magnitude of Bayes factors, to evaluate the strength of the comparison, the factor of N should be left in.

In order to use the Bayes factors to select the best model, we need to interpret the numerical value of the Bayes factors. Interpreting the magnitude of Bayes Factors heuristically is commonly done by considering the ratio as an odds ratio, e.g., odds of 20 to 1 in favor of the model in the numerator corresponds to a $BF = 20$, or, using natural logs, $\log BF = 3$. This choice of numerical cutoffs for the Bayes factors directly corresponds to a choice of prior probability to place on each model. This can be seen in equation 5 since the model with higher posterior odds is determined by the Bayes Factor times prior odds. In Table 1 we show a typical convention for interpreting Bayes Factors, with their suggested labels. This convention was developed by Jeffrey’s [23] and has been frequently applied in the literature.

To summarize, our method works as follows. For a given end-host computer, we compute $\log BF_{EP,P}$ as in Equation (8). The convention in Table 1 suggests that if the $\log BF_{EP,P}$ is larger than 3, then the EP model should be selected. If it less were 3 (i.e. the models are indistinguishable), then the P model should be selected

¹The Laplace approximation is known in the Physics literature as the *saddlepoint approximation*. [27]

Table 1: Interpretation of Bayes Factor strengths

Odds	$\log_{10}(BF)$	$\log(BF)$	Strength of comparison
20:1	1.3	3	“substantial”
100:1	2	4.5	“strong”
1000:1	3	7	“decisive”

as a final choice since it has fewer parameters. This convention is based upon a typical amount of data, and in our case we have a larger than usual dataset size. We thus increase the threshold to 10, which increases the odds that the best model has been selected. Put alternatively, this gives us more confidence to reject the pure Pareto model. If the EP model is selected, then we compute $\log BF_{EEP,EP}$. Again, if this factor is above 10, then EEP is selected, otherwise the final choice is EP.

5. VALIDATION

In this section, we validate our model selection and parameter estimation methods. In order to make comparisons with ‘ground truth’, we use synthetic data so that we may know the true value of the parameters of the generating data, versus the model parameters that our method estimates. Since initially it is just as likely that any of the 3 models is best for a given user, we need to validate that our method is capable of selecting any of the three models. First we use these models to generate samples of network traffic, which are in turn fed into our MLE optimizer that computes the BIC value for the fit along with estimates of $(\hat{\alpha}, \hat{m}, \hat{\lambda})$ for the EP model, $(\hat{\alpha}, \hat{m}_1, \hat{\lambda}_1, \hat{m}_2, \hat{\lambda}_2)$ for the EEP model, and simply $\hat{\alpha}$ for the P-model.

In this way we can compare our estimates with the true parameter values to gauge the accuracy of the method. In Fig. 2 we show the MLE method’s distribution of α estimates for EP models on EP sample data. The eight synthetic test values of α are indicated on the top of the plot. The box plot indicates the range of α ’s estimated by two different methods. For each test value, we ran 100 tests of 10000 samples each, generated from the EP model. The 100 tests spanned λ values from 0.1 to 0.3 and $m_1 = 0.5$. We see that the range of $\hat{\alpha}$ ’s in the columns subtitled “MLE” is almost always within a few percent of the true value as shown by the dotted horizontal line. This validates that our MLE optimization algorithm for the EP mixture model converges to an accurate value on data when run on simulated EP data.

In comparison, in [9], the authors develop a method called a ‘scale estimator’ for estimating the α parameter of heavy-tail distributions, based on a scaling property of sums of heavy-tailed random variables. With a publicly available implementation, called *aest*, this tool has been widely used in other research efforts, (e.g., [30]). An attractive property of this estimator is that it is nonparametric and easy to apply. This method is primarily used for the tails of α -stable distributions, and tries to handle the usual problem of identifying where the tail begins by carefully selecting the data used in computing the tail estimate. Since we know the ground truth, in these validation tests, we can compare the results produced by this tail parameter estimator with our EP model estimates. We restrict our comparison to $1 < \alpha < 2$, the range over which both methods apply. The $\hat{\alpha}$ values boxplots obtained via the AEST test are paired with the MLE boxplots on the same data in each panel of the figure. Interestingly we see that the AEST $\hat{\alpha}$ ’s have a higher variance and tend to be biased. In some sense, this is not surprising as the authors published results [9] acknowledge similar estimate variances when the underlying distribution is Pareto. Similarly the bias may well be due to the exponential component bleeding into

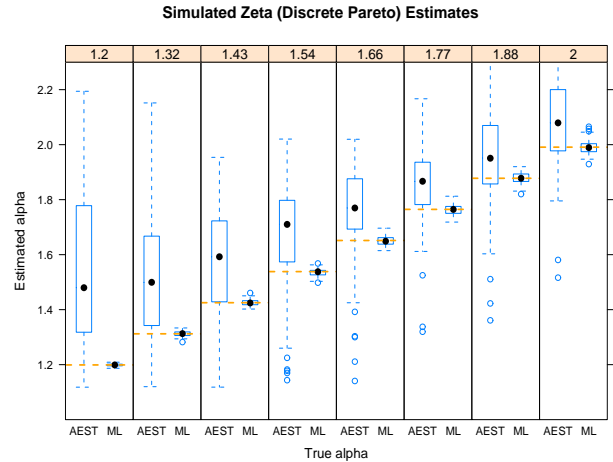


Figure 2: Comparing generated model parameters with estimated parameters for AEST and MLE methods.

the tail estimate. The advantage of AEST is its simplicity and lack of assumptions about the form of the overall distribution. AEST is also less computationally intensive, but since as a tail estimator it disregards part of the data, it cannot achieve the same statistical efficiency as the MLE estimator. However, since we need both an estimate conditioned on the full dataset for model selection and are willing to pay the computational price of direct optimization of the likelihood function, we elect to use the more heavy-handed, and more accurate estimate produced from an MLE.

Next we confirm that our model selection mechanism does indeed select the right model by comparing Bayes Factors for pairwise comparison of models run against simulated data from one of the models being compared. We ran these tests over a range of sample sizes, from 500 to 20,000 points, in the style of an empirical “design of experiments” to find what sample sizes were necessary to show adequate model selection results. For each test scenario with a given number of samples, we computed Bayes Factors over 100 test instances. We ran the following 3 test comparisons. In the first test, we generated samples from the EP model and computed the Bayes Factor for the EP versus the P models (hypotheses). Clearly, if the methodology works correctly it should prefer the true EP model. In the second test, we again considered EP data as the “truth,” and asked our method to select between EEP and EP as candidate models. For the third test, we instead generated EEP data, and used the method to choose again between EEP and EP as candidate models.

In Table 2, we summarize the ability of our model selection method to distinguish the 3 hypotheses. For each test, we state the minimum number of samples that were needed to achieve a conventional level of Bayes Factor, as noted by the terms *substantial*, *strong*, or *definite* corresponding to the conventions in Table 1. For the first two tests we list two results, showing how with additional samples we can get stonger results. For the first test, the P model is clearly distinguished and it doesn’t take many samples to “substantially” reject it. Then with as few as 5,000 samples, we can “definitely” prefer the EP model, underlining that the test between EP and P models is powerful. For the second test, we again see that a fairly small number of samples is sufficient to make “substantially” the right choice. However, getting to the next level of a “strong” preference for EP requires many samples (around 10,000). This is indicative of the similarity between the two models, suggested by

Truth	Model Choice:	Min Number Samples	\log_{10} BF strength
EP	EP vs. P	1000 5000	substantial definite
EP	EEP vs. EP	1000 10,000	substantial strong
EEP	EEP vs. EP	9000	substantial

Table 2: Confidence in model selection

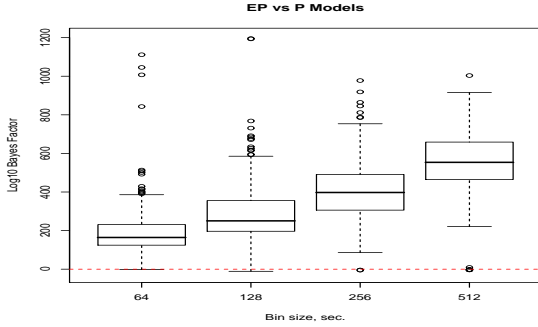


Figure 3: Boxplot of BIC comparison for Pareto vs. 2-component Mixture Model.

the qualitatively similar shapes of EP versus EEP. All in all with sufficient data we pick the right model because the EEP model incurs a penalty for extra complexity (recall the penalty term in the BIC factor) even though the EEP model subsumes the EP model. The third case, when the EEP model is true, is the hardest one to discriminate. Here when there were less than 9000 samples, EEP cannot be “substantially” distinguished from the EP. In summary, we see that eventually our method accurately selects the correct model—as long as enough samples are provided, but the power of the different pair-wise tests differs greatly. Specifically the EP vs. EEP choice is harder to make than the EP vs. P choice.

We reveal in the next section that in practice the Bayes Factors computed on our data have values ranging in the hundreds, with sample sizes in the thousands and tens of thousands. Thus the ability to discriminate among models on real data appears stronger than this validation exercise would suggest. This may mean that the “true” models don’t lie between the hypotheses, but much closer to one hypothesis than the other. This is reason to believe that requiring samples on the order of a few thousand (or at most 10,000) is a fairly light requirement for model selection in our domain.

6. RESULTS

In this section, we used our methodology to select the best model for each of our 270 users. We then make some observations about our users based upon the selected models and model parameters.

6.1 Choice of Models

First we compare the P and EP models for each user. In Fig. 3 we plot the log of the Bayes Factor (or difference in BICs) of the two models. The x-axis labels indicate the bin size that was used when the models were computed. For each bin size, we computed the $\log BF_{EP,P}$ for each user. For each bin size, we use box plots to show the distribution of the Bayes factors over all the users. We can see that for nearly all users we can select the two component EP model as ‘definite’, according to convention on BIC

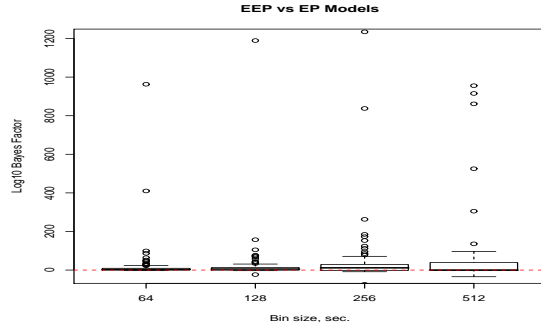


Figure 4: Comparison of EP and EEP models

strength as indicated in Table 2. There are a very small number of users whose whose $\log BF_{EP,P}$ was near zero. Recall from Section 4 that when two models are considered indistinguishable, the model with fewer parameters is selected. The methodology selects a Pareto-only model for roughly a dozen of our endhost machines. Not only is the two component mixture model EP preferred for all the other users, but it is strongly preferred as evidenced by the high Bayes factor values. We observe a small trend here in that as the bin sizes increase, the log Bayes factor ratio gets larger. This empirical observation indicates that for larger bin sizes, the exponential component plays an increasingly dominant role. Although not shown here on graphs, we also observed that as the bin sizes increase, the median mixing fraction m increases. This corroborates the observation that the exponential plays an increasing role for larger bin sizes.

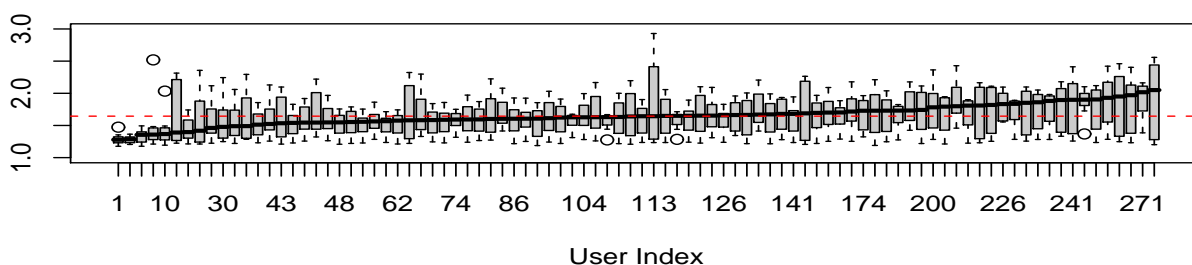
Next we compare the EP and EEP models. Fig. 4 plots the distribution (as a boxplot) of the Bayes factors over all users, for each of 4 bin sizes. Interestingly, we see that at bin sizes of 64 and 128, the Bayes factors are close to zero for the majority of the users. Since the two models are fairly indistinguishable here, we again select the model of lower complexity, namely EP for nearly all the users. (There are a few outliers that would elect EEP). At larger bin sizes, we do see some users for whom the EEP model is selected. Overall, our method assigns the EEP model to roughly 30% of the users and the EP model to the remaining 70%.

Fig. 5 shows which model is selected by the methodology for all of our endhost machines. Overall we see that only a handful of users are given the Pareto-only model, and between 15%-40% of user machines are best modeled by EEP (depending upon the bin size). Overall, the EP model is selected for 50-85% of the users, again depending upon the bin size. We conclude two things from this section. First, the flexibility we have built into our methodology is important and needed because the best model for one endhost is not necessarily the same for another endhost. Second, for the majority of the endhosts, the mixture model consisting on one exponential and one Pareto is clearly the preferred model.

6.2 User Behavior

As indicated in Section 2, there is a growing interest in understanding the range of variation of user behavior. We now look at some of the model details to explore the range of parameters selected across users, and the amount of mixing between the two model components. We computed an EP model for all our users, and show the α and λ model parameters for each user in Fig. 6. The users are ordered along the x-axis, in terms of increasing value of median α . For each user, the upper plot shows a boxplot describing the range of α ’s selected across eight bin sizes. This plot indicates

Zeta (Discrete Pareto) Alpha Parameter



Exponential Lambda Parameter

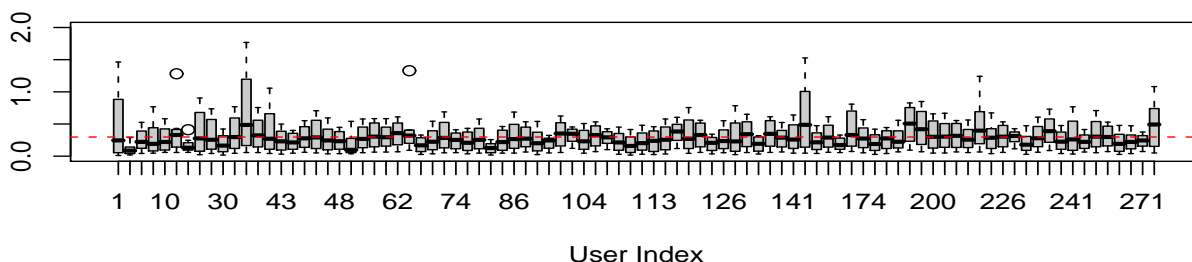


Figure 6: Distribution of (α, λ) across users. Sorted in increasing order or average α . The dotted red line is the mean α over the set of users.

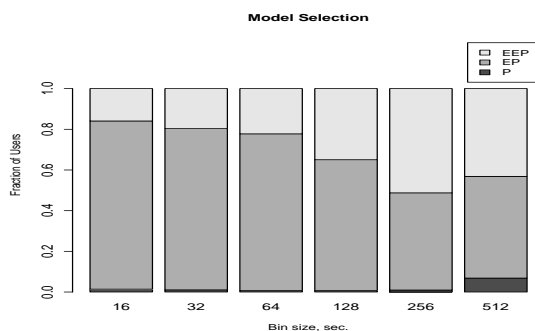


Figure 5: Choice of models based on Bayes Factors for different users. Each bar represents the same users with a different binning time window

that the particular value of α selected by any individual user can vary substantially depending upon the bin size. We also see that the median values of α range from 1.3 to 2.3 across the users. The slope of the tail across users can thus vary substantially in terms of how "heavy" it is. Those with α 's approaching 2 have lighter tails than those whose α is closer to 1. The lower plot shows the value of λ selected by the users. Here we see that there is little variation in the value of λ chosen.

In earlier Internet traffic modeling studies, the α parameter lies between 1 and 2. Here we see α 's larger than two. There is nothing inherent in the data, nor in Pareto distributions that limits the $\alpha < 2$ [6]. Distributions with $\alpha \geq 2$, have finite variance. The focus on $\alpha < 2$ has come from the interest in studying distributions with infinite variance. Similarly, when $\alpha < 3$, the third moment is infinite.

To see more carefully the details of the distribution of α 's over users, we provide a histogram for the selected α values for each user at a bin size of 64 in Fig. 7. We see that user's have very different properties in terms of the heaviness of the tail of the distribution. Roughly 1/6 of our users, have $\alpha < 1.5$ implying a fairly heavy tail. Most of our users have α 's around 1.6 or 1.7. It is interesting to see that we do have a small number users (4) with $\alpha > 2$ indicating a finite second moment. This range of α 's indicates that there is a great deal of diversity in shape of the tails across users.

We now look more closely at how the users mix the two components of the model. A value of m close to 0 implies that the model is dominated by the exponential distribution, (when $m = 0$ there is no Pareto component in the model). Similarly when m is close to 1 the Pareto component dominates the behavior of the model ($m = 1$ indicates there is no exponential component). The m parameter is considered a *soft* model selection factor because of its ability to

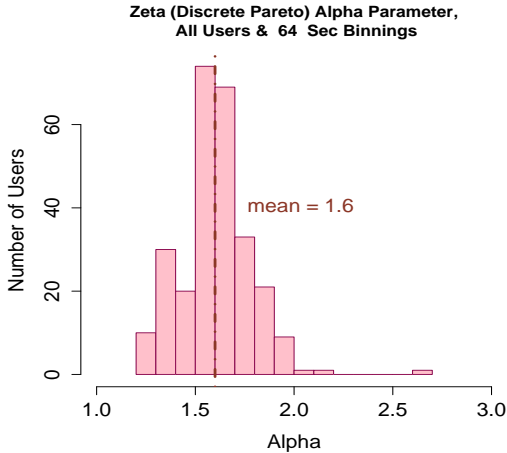


Figure 7: Histogram of estimated α values across users at a 64 second bin size.

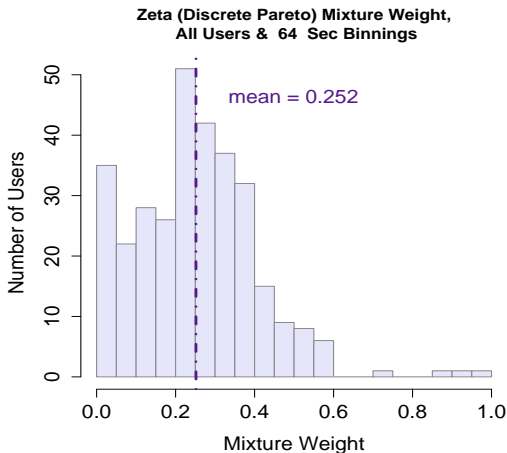


Figure 8: Histogram of estimated m_{α} across users at a 64 second bin size.

indicate the strength of each component of the distribution. The MLE estimates determine m from the data, which is why it can be viewed as a soft model selection factor. To see the range of m values chosen across our users, we provide a histogram of this mixing factor in Fig. 8. The frequency on the y-axis denotes the number of users whose m parameter is that indicated on the x-axis. We see that only 3 users picked an m very close to 1, indicating that the pure Pareto model suites practically none of our users—in agreement with the Bayes Factors conclusions. Most of the users have an m parameter less than 0.4, and roughly half of our users had $m < 0.25$ indicating the dominance of the exponential component in the model. The m values are fairly well spread across the range 0 to 0.5 (roughly). We can also interpret this range of m as a indication of user diversity, in that their mixing fractions differ substantially.

7. DISCUSSION

There are a number of scenarios in which the models we develop in this paper have application.

7.1 Synthetic Traffic Generation

model	OFF	ON
P	7.7%	6.4%
EP	92%	84%
EEP	0.3%	9.6%

Table 3: Percentage of users that selected P, EP and EEP models for OFF periods. Same for ON periods.

One of the uses of statistical traffic models is in synthetic traffic generation. There has been a considerable interest over the past few years in developing traffic generators, e.g., for use in testing protocols and load balancing schemes (see [1] for a comprehensive list). Typically such tools generate traffic such as one would see near routers or gateways, i.e., aggregated UDP streams and TCP streams simulating a collection of applications. However, it is not necessarily the case that *all* traffic from a given user is modeled in such cases. Our models fill this gap, describing live traces of user traffic directly at endhosts, and can be used in testing scenarios when individual user traffic is needed.

The results we report in this paper lend themselves naturally to a traffic generation methodology based on a two-state model. We think of the flows in and out of an endhost as being either “ON” or “OFF.” The endhost alternates between ON and OFF states; duration in the ON state is governed by a particular distribution, as is duration in the OFF state. While in the ON state, the number of network flows per bin is given by one of the distributions studied in the previous Section. The OFF (“network-idle”) state captures episodes when the machine is powered on but does not generate any traffic.

Note that this model can be used to generate traffic for an individual user, or by aggregating multiple instances of the model, one can generate traffic for a population of users. When multiple users are involved, the parameters of the flow arrival process across users, as well as the specific model (P, EP, or EEP) for each user, may be varied as discussed in the previous Section. To populate such a population model, one could gather tables that capture many triples (λ, m, α) , reflecting a broad range of values and legitimate pairings of these parameters. This could be used via distribution sampling to generate flow-level samples. More generally, this triple could itself be modeled by a multi-variate distribution, or via a hierarchical Bayes model.

To complete such an ON/OFF traffic generator, one needs models for the ON and OFF periods. Hence, we used our data to develop those models, using the same methodology we applied to connection counts. For each endhost, we used our methodology to select either P, EP or EEP to model the length of each users’ ON periods and OFF periods. Note that one model (e.g., EP) can be selected to model the length of the OFF period, while a different model (e.g., EEP) can be selected to model the ON period length. We calibrated ON-OFF models for all of our endhosts; summary findings are given in Table 7.1. The table shows the percentage of users that choose the P, EP or EEP model as the best to describe the length of both ON and network-idle periods. Once again, the EP model is most often the best; we see for example, that for 92% of endhosts, the length of network-idle periods is best captured by the EP model.

Our calibration results in a choice of model, as well as values of the corresponding model parameters, for each endhost. In Table 7.1 we list values for each model parameter averaged across users. For each state, OFF and ON, we provide the parameters of

model	α	λ_1	λ_2	m_1	m_2
OFF					
P	2.06				
EP	2.35	0.77		0.16	
EPP	2.24	1.27	0.75	0.13	0.12
ON					
P	1.89				
EP	2.15	0.47		0.36	
EPP	2.7	0.72	0.43	0.16	0.46

Table 4: Mean values of ON-OFF model parameters

the model, when P is selected, when EP is selected, etc. To interpret the data in Table 7.1 consider an example. When the EP model is selected to model the OFF period, then the values of the 3 parameters for this model, (α, λ_1, m_1) , are given in the Table. Each value stated is an average computed over all endhost machine that selected that model for the given state. (Space does not permit giving a more detailed characterization of the calibrated ON-OFF models, but the Table gives some examples that may be used to generate synthetic traffic.)

7.2 Traffic Composition

The traffic models described this far are high level models that are agnostic as to the particular kinds of applications or services present in the traffic. An interesting future direction of interest is in uncovering the generative models behind the traffic being generated. Are there particular applications and ports that tend to occur more often in the exponential component of the distribution, or the pareto component? Are there particular types of traffic that are generated by human interaction, or by the background processes on the host? While a detailed analysis of such questions is outside the scope of this paper, in this section we present some initial findings towards these questions.

We used our traces to see which applications are being used during each of the two behavioral regions, 'exponential' and 'tail'. We can soft-cluster the bins in each user trace (independently), as belonging to the 'exponential' or 'tail' region of the model by comparing the connection counts against our threshold that marks the start of the tail. This clustering (or labeling) indicates which component of the model is dominant in that window of time. Using our keyboard and mouse click data to associate with each bin a flag that indicates if the user was active or idle in this time window. We use a simple and conservative heuristic: the user is considered idle in a time window if there was *no* recorded user activity in the window, and active otherwise.

From our dataset, we extracted the top 24 ports ranked by total count across all the users and further semantically grouped them into a smaller set of 9 traffic categories (loosely, applications) of interest. For instance, tcp traffic on ports 80, 443 and 8080 was grouped into a "web traffic" category; we noticed dns traffic on both tcp and udp, which was combined into a single "dns traffic" category. We present results for these 9 categories.

Fig. 9(a) plots the flow counts for a particular traffic category as observed in bins falling in the exponential (or pareto) part of the mixture model. The counts are normalized by dividing the counts by the total flows observed in the exponential (or pareto) bins, respectively. For each application category, the bar on the left refers to the exponential bins, whereas the bar on the right describes the pareto labeled bins. We see that the traffic in the pareto tails is dominated by four traffic categories, DNS, Web, ICMP and Bixfix.

Bixfix is an enterprise application that automatically manages software updates. Large ICMP bursts in our enterprise are known to occur due to activities that scan multiple servers to find the closest one for a download. The behavior during so called "exponential" bins (windows of time) appears to be driven by all 8 categories shown, with Web, DNS and ICMP being the primary drivers. One may postulate that the Web and DNS traffic (clearly these two go together) is primarily human-triggered activity. ICMP is present to a roughly equal degree in both the exponential body and the pareto tail; the implication being that the ICMP "bursts" vary a great deal in size.

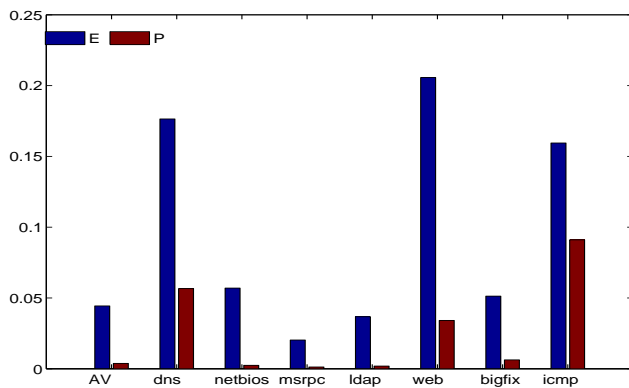
Fig. 9(b) plots the flow counts for a particular traffic category as it is observed in bins where the user is idle and when the user is active. Again, the counts were normalized by the total flow counts in each class. In this breakdown of the traffic, we see that most of the traffic categories examined are present in equal measure whether the user is idle or active. On the one hand, the existence of a fair amount of heavy-tailed traffic during user-idle periods is somewhat surprising because it opposes findings from other heavy-tailed research studies claiming that the reasons behind the heavy-tailed traffic is due to user behavior. On the other hand, it makes sense when you consider modern day practices for configuring enterprise clients. Such clients come pre-configured with a great deal of software for security (AV engines that get updated multiple times a day), monitoring, software compliance checking and software updating. These enterprise applications are autonomous and generate traffic whether or not the user is active. Web traffic is the only category that differs substantially between user-active and user-idle time periods. The web traffic during user-idle periods may reflect web content that is refreshed aggressively, and also asynchronous (eg. AJAX) style applications.

While the results presented here are extremely preliminary, there is evidence that points to specific applications (and traffic types) contributing more to one part of the mixture model distribution, and that there are traffic types that have a strong dependence on the state of the user. We hope to explore these in future work.

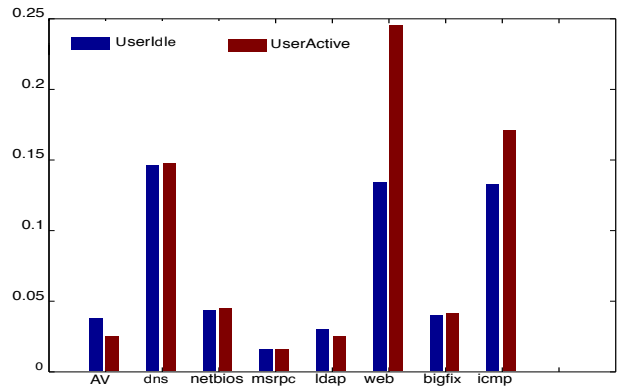
8. CONCLUSION

In this paper we set out to model flow traffic as generated by endhost machines such as enterprise employee laptops. We employ mixture models based on a convex combination of component distributions with both heavy and light-tails. We approach the modeling problem as a model selection problem rather than a goodness-of-fit test. Our methodology selects the best model for an endhost by considering a family of 3 models and doing pairwise comparisons to pick the best one. We employ the Bayes factor, based on the Bayesian Information Criteria (BIC), for these comparisons. To the best of our knowledge, this is the first paper to study heavy tails of data collected directly on endhosts, and is the first to employ a model selection approach.

We apply our methodology to data collected from 270 enterprise users, and uncover a variety of salient findings about this data. Primarily, we find that for the vast majority of users, the methodology selects the EP model. Although there are some users best modeled by EEP, and few by P. This shows the importance of a method that users a family of distributions and does not presuppose a single distribution model for flow traffic. We learn that our enterprise user population contains a great deal of diversity; not only do different users need different models, but some are heavy-tailed and others not. We observe a wide range of values for the tail slope and mixing fraction in our models. We also showed how our model can be applied to find ON-OFF models capturing network-active and network-idle periods. Such models are an important step in syn-



(a) Flow counts across bins marked 'exp' and 'pareto'



(b) Flow counts across bins where user was idle or active

thetic traffic generation. We take an initial glance deeper into the network traffic and see hints that a small number of protocols and applications may be responsible for the observed heavy tail behavior. We also see the presence of heavy-tailed traffic when users are idle indicating that the flows comes from machine-generated traffic (such as enterprise applications and chatty protocols). In the future we plan to further explore the generative models behind the traffic patterns we observed herein.

9. REFERENCES

- [1] <http://www.grid.unina.it/software/ITG/link.php>.
- [2] BARFORD, P., BESTAVROS, A., BRADLEY, A., AND CROVELLA, M. Changes in Web client access patterns: Characteristics and caching implications. *World Wide Web* (1999).
- [3] BARFORD, P., AND CROVELLA, M. E. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of Performance '98/SIGMETRICS '98* (July 1998), pp. 151–160. Software for Surge is available from Mark Crovella's home page.
- [4] BHARMAN, D., CHANDRASHEKAR, J., TAFT, N., FALOUTSOS, M., HUANG, L., AND GIROIRE, F. Debating IT Monoculture for End Host Intrusion Detection. *ACM Sigcomm Workshop on Research in Enterprise Networks* (2009).
- [5] BRESLAU, L., CUE, P., CAO, P., FAN, L., PHILLIPS, G., AND SHENKER, S. Web caching and zipf-like distributions: Evidence and implications. In *IN INFOCOM* (1999), pp. 126–134.
- [6] CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. J. Power-law distributions in empirical data. *SIAM Review*. To appear (2009).
- [7] CROVELLA, M., AND KRISHNAMURTHY, B. *Internet Measurement*. John Wiley & Sons, West Sussex, England, 2006.
- [8] CROVELLA, M. E., AND BESTAVROS, A. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. on Networking* 5.
- [9] CROVELLA, M. E., AND TAQQU, M. S. Estimating the heavy tail index from scaling properties. In *Methodology and Computing in Applied Probability* (1999), pp. 55–79.
- [10] CUNHA, C. A., BESTAVROS, A., AND CROVELLA, M. E. Characteristics of WWW client-based traces. Tech. Rep. TR-95-010, Boston University Department of Computer Science, Apr. 1995. Revised July 18, 1995.
- [11] EVERITT, B. S., AND HAND, D. J. *Finite Mixture Distributions*. Chapman and Hall, London, 1981.
- [12] FELDMANN, A. *Self-Similar Network Traffic and Performance Evaluation. Chapter 2: Characteristics of TCP Connection Arrivals*. John Wiley & Sons, New York, NY, 2002.
- [13] FELDMANN, A., GREENBERG, A., LUND, C., REINGOLD, N., REXFORD, J., AND TRUE, F. Deriving traffic demands for operational ip networks: Methodology and experience. *IEEE/ACM Transactions on Networking* 9 (2001), 265–279.
- [14] FELDMANN, A., AND WHITT, W. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In *Proceedings of IEEE INFOCOM'97* (April 1997).
- [15] G. TAN, M. POLETO, J. G., AND KAASHOEK, F. Role Classification of Hosts within Enterprise Networks Based on Connection Patterns. *USENIX Annual Technical Conference* (2003).
- [16] GIROIRE, F., CHANDRASHEKAR, J., IANNACCONE, G., PAPAGIANNAKI, K., SCHOOLER, E., AND TAFT, N. The Cubicle Vs. The Coffee Shop: Behaviora Modes in Enterprise End-Users. *Passive and Active Measurement Workshop (PAM)* (2008).
- [17] ISDAL, T., PIATEK, M., KRISHNAMURTHY, A., AND ANDERSON, T. Leveraging BitTorrent for End Host Measurements. *Passive and Active Measurement Workshop (PAM)* (2007).
- [18] J. M. MARIN, K. M., AND ROBERT, C. Bayesian modelling and inference on mixtures of distributions. Tech. rep., CEREMADE, Universite Paris Dauphine, February 2004.
- [19] JOO, Y., RIBEIRO, V., FELDMANN, A., GILBERT, A. C., AND WILLINGER, W. TCP/IP traffic dynamics and network performance: a lesson in workload modeling, flow control, and trace-driven simulations. *SIGCOMM Comput. Commun. Rev.* 31, 2 (2001).
- [20] JORDAN, M. I., AND JACOBS, R. A. Hierarchical mixtures of experts and the em algorithm. *Neural Computation* 6 (1994), 181–214.
- [21] JÖRG WALLERICH, HOLGER DREGER, A. F. B. K., AND WILLINGER, W. A Methodology for Studying Persistency Aspects of Internet Flows. *SIGCOMM CCR* (2005).
- [22] KARAGIANNIS, T., MORTIER, R., AND ROWSTRON, A. Network exception handlers: Host-network control in enterprise networks. *ACM SIGCOMM* (2008).
- [23] KASS, R. E., AND RAFTERY, A. E. Bayes factors. *Journal of the American Statistical Association* 90, 430 (1995), 773–795.
- [24] LELAND, W. E., TAQQ, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of Ethernet traffic. In *ACM SIGCOMM* (San Francisco, California, 1993), D. P. Sidhu, Ed., pp. 183–193.
- [25] LI, L., ALDERSON, D., WILLINGER, W., AND DOYLE, J. C. A First-Principles Approach to Understanding the Internet's Router-level Topology. *Proc. ACM SIGCOMM* (2004).
- [26] LUO, S., LI, J., PARK, K., AND LEVY, R. Exploiting Heavy-Tailed Statistics for Predictable QoS Routing in Ad-Hoc Wireless Networks. *IEEE Infocom* (2008).
- [27] MACKAY, D. J. C. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2003.
- [28] MITZENMACHER, M. Editorial: The future of power law research. *Internet Mathematics* 2, 4 (2006), 525–534.
- [29] NEWMAN, M. E. J. Power laws, pareto distributions and zipf's law. *Contemporary Physics* 46, 5 (2005), p323 – 351.
- [30] PAPAGIANNAKI, K., TAFT, N., AND DIOT, C. Impact of flow dynamics on traffic engineering design principles. In *Proceedings of IEEE Infocom, Hong Kong, March 2004* (2004).
- [31] PAXSON, V. Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM Trans. Netw.* 2, 4 (1994), 316–336.

- [32] PAXSON, V. Bro: A system for detecting network intruders in real-time. *Computer Networks* (1999).
- [33] PAXSON, V., AND FLOYD, S. Wide-area traffic: the failure of poisson modeling. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications* (New York, NY, USA, 1994), ACM, pp. 257–268.
- [34] SIMPSON, C., REDDY, D., AND RILEY, G. Empirical Models of TCP and UDP End-User Network Traffic from NETI@home Data Analysis. *Principles of Advanced and Distributed Simulation PADS* (2006).
- [35] VAN DER VAART, A. W. *Asymptotic Statistics (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, June 2000.
- [36] VISHWANATH, K. V., AND VAHDAT, A. Realistic and responsive network traffic generation. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2006), ACM, pp. 111–122.
- [37] WALLERICH, J., AND FELDMANN, A. Capturing the variability of internet flows across time. In *Proceedings of the INFOCOM 2006, 25th IEEE International Conference on Computer Communications, 9th IEEE Global Internet Symposium* (April 2006), pp. 1–6.
- [38] WILLINGER, W., TAQQU, M. S., SHERMAN, R., AND WILSON, D. V. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Trans. Netw.* 5, 1 (1997), 71–86.
- [39] WRIGHT, S. J. *Primal-Dual Interior-Point Methods*. SIAM Publications, 1997.
- [40] ZHANG, Y., BRESLAU, L., PAXSON, V., AND SHENKER, S. On the characteristics and origins of internet flow rates. In *ACM SIGCOMM* (2002), pp. 309–322.