An Atomic Theory of Flow Behavior

Stefan Karpinski, Elizabeth M. Belding, Kevin C. Almeroth
Department of Computer Science
University of California, Santa Barbara
{*sgk,ebelding,almeroth*}@*cs.ucsb.edu*

Abstract— **We propose an entirely new approach to understanding and analyzing the behavior of flows in packet networks. The essential concept of this new approach is to find atomic units of time and size behavior in traces of network flows. While the nonparametric statistical techniques for extracting these basic behavioral units are complex, the end result is quite simple: the units provide an alphabet for flow behavior. From a finite set of behavioral units, an infinite variety of actual behaviors can be composed. The space of behaviors naturally becomes a vector space generated by these atomic units of behavior. We use numerical linear algebra techniques to demonstrate useful and immediate applications of this theory to real-time traffic analysis, anomaly and attack detection, as well as workload generation for wireless experiments.**

I. Introduction

The trouble with trying to understand or model behavioral patterns in packet networks is that beyond the packets themselves there is no inherent behavioral structure. There are flows of packets with the same source and destination IP addresses and TCP/UDP port numbers, and sessions of flows belonging to the same host, but these imply only very limited behavioral similarity. Each one has its own unique sequence of packet sizes and inter-packet intervals with no obvious relation to each other. Without fundamentally behavioral elements of structure, traffic traces are just "packet soup," so to speak. Accordingly, traditional approaches to traffic analysis have either focused on aggregate traffic measures or categorized flows by well-known port numbers and application types. Some fiendishly clever techniques have been proposed to tease out the applications types underlying traffic found in network traces [1]. Without an inherently behavioral theory of network traffic, however, we believe that insight into the fine structure of network behavior is ultimately very limited.

We propose to turn this problem on its head by providing an atomic behavioral theory for network traffic. We begin with the concept of a packet flow as a natural starting point and define a "flowlet" as a segment of a flow with statistically consistent packet size or inter-packet interval behavior. Common flowlet behaviors are extracted from a body of traffic traces, using statistical clustering. These clusters of similarly behaved flowlets provide the atomic units of flow behavior: by mixing basic behaviors from a finite collection of flowlet clusters in varying proportions, an infinite variety of flow behaviors emerge. It becomes natural to view the space of flow behaviors as a vector space of linear combinations of flowlet clusters.

To demonstrate the viability and utility of this atomic theory of flow behavior, we apply standard numerical linear algebra techniques to the resulting vector space of flow behaviors. Principal component analysis (PCA) allows us to reduce the dimensionality of the vector space of flow behaviors while preserving the vast majority (99%) of the variability in the original data. PCA allows us to represent flow behaviors concisely in only eight dimensions ($\mathbb{R}^8$). Moreover, this representation has several desirable properties. The eight coordinates of flows mapped into this space are linearly uncorrelated (nonlinear dependencies, however, remain). The first dimension of the transformed data captures the most important differences between flows; the second dimensions the next most important, and so on. The first two or three dimensions, thus, are ideal for visualization of behavioral differences. Finally, the dimensions are naturally scaled so that the standard Euclidean distance provides a good measure of the behavioral (dis)similarity between flows. As a result, standard multi-dimensional analysis and modeling techniques, such as k-means clustering, can be applied directly to the transformed flow behaviors.

We present two useful, immediate applications of this new theory of flow behavior. First, once a body of traffic has been analyzed and PCA-transformation matrices computed, new flows can be mapped into the flow-behavior space using only fast matrix and vector computations with pre-computed matrices. This allows the possibility of completely real-time traffic analysis and visualization. Second, since the coordinates of the PCA-transformed flow behaviors are uncorrelated and exhibit limited non-linear dependencies, we can roughly model them using a multivariate normal distribution. Random deviates from this distribution can be mapped back into actual flow behaviors, allowing the generation of an entire network's worth of heterogeneous synthetic flow behaviors with the realistic intra-flow and inter-flow behavior matching the original trace traffic. This ability is of particular importance for experimental wireless research, where it has been shown that using standard, naïve traffic models, such as the uniform constant bit-rate (CBR) flows, severely distorts important performance metrics at all levels of the network [2], [3].

II. Motivation & Related Work

In Internet traffic analysis, the detailed structure of workload patterns in local-area networks (LANs) is of limited interest. Capacity has become so cheap and plentiful in wired LANs that workload details are simply irrelevant in the face massive over-provisioning. Wireless networks, however, are fundamentally different: the entire medium is at the "edge" of the network and the most basic resources of bandwidth and power are severely limited, with no permanent relief in sight. Thus, over-provisioning is simply not an option, and the fine-grained details of traffic patterns have been shown to have a dramatic impact on network performance [2], [3]. As a result, modeling and analysis of wireless LAN (WLAN) traffic has recently become a hot topic in the wireless community, and a significant body of high-quality research on this subject has been produced [2]–[7].

Most of the recent surge of research in WLAN traffic has taken a top-down approach, using parametric models to reproduce high-level statistical characteristics of observed workload behavior [4]–[7]. In particular, Herández-Campos *et al.* [5] showed convincingly that the following models are applicable to WLAN traffic: user arrivals (sessions) follow a time-varying Poisson model; the number of flows per session follows a BiPareto distribution; the sizes of flows follow a BiPareto distribution; the intervals between the initiations of flows within each session follow a Lognormal distribution. These models provide an excellent and convincing high-level overview of the behavior of users and applications in WLANs.

The methodology for generating synthetic workload remains incomplete, however. While the high-level scaffolding for producing WLAN traffic exists, the low-level behavior of flows is neither understood nor reproducible. The common practice in workload generation is to use a uniform constant bit-rate (CBR) model for the packet-level behavior of flows: all flows have the same number of packets, all packets have identical size, and the intervals between packets in each flow are of a single, fixed duration. Karpinski *et al.* [3], however, showed that all types of CBR packet behavior models drastically distort important performance metrics, and thus fail the litmus test for realism. Distorted performance resulted even when CBR was applied with high-level behavior taken directly from traces *and* using accurate packet count, average packet size, and average inter-packet interval for each flow [2].

Using variable bit-rate (VBR) flows is a common elaboration upon the CBR flow model. In VBR models, the packets sizes and the inter-packet intervals are each randomly sampled from pre-specified, independent, identically distributed (i.i.d.) distributions. One of the most significant results of [2] is that if high-level workload is realistic, then an i.i.d. VBR model for flow behavior accurately reproduces network performance—with one *very* major caveat: each flow must have its own unique, realistic signature of packet size and inter-packet interval distributions. Producing a realistic cross-network collection of these distributions is, to date, an unsolved problem. The size and interval distributions used in [2] were empirical estimates, taken directly from the observed behavior of each flow in the original trace. This result has a very important implication for the present work: *to characterize flow behavior, it is sufficient to know each flow's distribution of packet sizes and inter-packet intervals*. It is precisely this problem of understanding, modeling, and recreating realistic collections of packet size and inter-packet interval distributions for an entire network that this paper addresses.

III. Methodology

Our goal is to provide an effective general methodology for analyzing and characterizing the space of packet size and inter-packet interval distributions found across the flows in WLAN trace traffic. In essence, we require a "distribution of distributions." However, size and interval distributions exist in spaces with very high dimensionality. In theory, both types of distributions live in infinite-dimensional Hilbert spaces. In practice, limits on possible values and quantization make the effective dimensions finite, but nonetheless too large for simple, direct analysis.

To reduce the dimensionality of distribution data, we use a series of techniques. First, we divide each flow's sequence of values—sizes or inter-packet intervals—into segments of consistent behavior, using time series change point detection. We call these segments of flows "flowlets." Second, we apply agglomerative clustering to find clusters of flowlets with mutually consistent behaviors. These flowlet clusters are the atomic units that allow us to systematically apply standard techniques from other fields to analyze and understand flow behaviors. Each flow's behavior can be expressed as some weighted sum of constituent flowlet clusters. Thus, the space of flow behaviors have naturally become a vector space generated by the flowlet clusters. In the final stage of processing we use principal component analysis (PCA) to reduce dimensionality of this vector space and isolate the most important aspects of behavior across the entire collection.

A. *Nonparametric Goodness-of-Fit Tests*

In the case of packet sizes and inter-packet intervals, there are no "clean" parametric models that capture the variety of behaviors seen in network traces. Therefore, we rely on nonparametric statistical procedures for our analysis. There are a number of nonparametric two-sample goodness-of-fit tests in the statistical literature. Such tests take two samples of scalar values and test the null hypothesis that the samples were drawn from the same (unknown) underlying distribution. Nonparametric tests do not make any assumptions about the shape or nature of the underlying distributions, other than that they are continuous and real-valued. Such tests can detect various differences in both location and shape of distributions. We use the Baumgartner-Weiß-Schindler (BWS) test [8] rather than the more well-known Kolmogorov-Smirnov (KS) or Cramér-von Mises (CvM) tests because: BWS exhibits significantly superior power in simulation; the exact distribution of the BWS test statistic for small sample sizes is already very close to its asymptotic distribution, which is untrue for KS and CvM.

B. *Flow Splitting*

The first analytical task is to split the packet size and inter-packet interval time series for each flow into segments with consistent statistical behavior. A great deal of classical statistical research has been done in the general area of time series change point detection [9]. These classical approaches, however, cannot be applied when the number of change points are unknown and, worse still, the underlying models for the stochastic processes are unknown. Work has been done in econometrics that allows a single change point to be discovered using nonparametric KS or CvM type significance tests [10]. Zheng *et al.* [11] use recursive iteration of this procedure with the Fligner-Policello (FP) test to detect change points in Internet path properties. Our technique is the same except that we use the BWS test rather than FP; the FP test detects only changes in central location, whereas BWS detects changes in both location and shape.[1] Allen *et al.* [12] use this procedure with the KS test to detect change points in time series of network bandwidth measurements. Space unfortunately does not permit us to reproduce the procedure here. The output of change point detection algorithm is a collection of "flowlets"—segments of flows where the packet size or inter-packet interval behaviors are statistically consistent. The flowlet splitting procedure is applied to intervals and sizes independently, so there are two separate collections of flowlets. Splitting and clustering of intervals and sizes are treated separately because [2] indicates that these can be modeled independently without loss of realism.

C. Flowlet Clustering

The second stage of our methodology takes the time and size flowlets produced by the first stage and clusters them into groups exhibiting similar behaviors. Again, there is an extensive body of work on classical clustering algorithms [13]. As before, however, the classical techniques cannot be applied because the data are fundamentally different from those found in classical problems. Our data are not vectors in a classical Euclidean space, but rather random samples from unknown, nonparametric distributions. Two-sample goodness-of-fit tests provide the ability to compare a pair of samples, but to the best of our knowledge there exists no prior art in statistics literature extending such tests to clustering collections of samples into similar groups. A technical note from management science [14], describes an interative clustering technique based on the KS test. Our procedure resembles the agglomerative version of their procedure. We utilize the BWS test rather than KS, however, for the same reasons as discussed in Section III-B. Our algorithm can be applied with any general two-sample nonparametric goodness-of-fit test, however, since the test is used only as a black-box generator of p-values.

The algorithm proceeds as follows. Initially, place each flowlet in its own cluster. Apply the goodness-of-fit test to each pair of clusters. Take the pair with the maximum p-value and merge them into a singler cluster, replacing the pair; the sample for the new cluster is the union of two samples. Repeat until the maximum p-value falls below the critical threshold ($\alpha = 0.05$). This indicates that the difference between all pairs of clusters is statistically significant, and no more agglomeration can be performed. The output of the algorithm is a mapping of flowlets to clusters such that the behaviors within each cluster are statistically insignificant and the differences between clusters are statistically significant.

The groups of flowlets produced by the clustering algorithm are fundamental "atoms" of behavior which can be mixed in different proportions to construct an unlimited variety of flow behaviors. Sufficiently realistic behavior, as defined in [3], can be reproduced by generating each flow's interval and size distributions as weighted sums of its constituent flowlet cluster distributions. This weighted combination accurately reproduces the original distributions of the flow, thereby preserving realism as shown in [2]. The weighting factor for each cluster distribution is just the fraction of packets in the flow from that particular cluster.

D. Principal Component Analysis

At this point, each flow's behavior can be expressed as a weighted sum of the flowlet cluster behaviors extracted in the first two stages of processing. This makes it natural to consider the individual flow behaviors as vectors in a space that is spanned by the flowlet clusters. We embed the clusters in $\mathbb{R}^n$ with $n = 200$ by converting each cluster's empirical cumulative distribution function CDF into a vector via quantization: time values are log-transformed; size values are square-root-transformed; transformed values are normalized and rounded into n bins.[2] The fraction of sample values falling at or below the ith bin becomes the ith coordinate of the CDF-vector for that sample.[3]

Before we can proceed further with numerical linear analysis, we must convert all cluster and flow data into vectors and matrices. Let $f, c_s, c_t \in \mathbb{N}$ be the number of flows, time clusters, and size clusters, respectively. Let $S \in \mathbb{R}^{n \times c_s}$ and $T \in \mathbb{R}^{n \times c_t}$ be the matrices whose columns are the CDF-vectors of the size and time clusters. Let $F \in \mathbb{R}^{f \times (c_s + c_t)}$ be a matrix with a row for each flow, and a column for each time and size cluster, and let F_{ij} be the fraction of the ith flow's values which belong to the jth cluster. This matrix expresses the behaviors of all flows as weighted mixtures of clusters. The matrices F, T, and S contain sufficient information to reconstruct size and time distributions for every flow.

The final processing stage uses principal component analysis (PCA) to reduce the dimensionality of these matrices without discarding essential information content. PCA takes a collection of vectors and finds a new orthogonal basis for the vector space with the following properties. The variance of the data projected onto the first coordinate is maximal; the projection of the data excluding that subspace onto the next coordinate is maximal as well, and so on. In this manner, the first coordinate of the PCA-transformed data expresses the most important aspects of the data, while the second coordinate expresses the next most important. Another property of

[1] Our "backward elimination" post-processing procedure differs slightly too: we eliminate the least significant change point candidate first and iterate until only significant ones remain; Zhang *et al.* eliminate insignificant candidates in sequential order. In cases where these procedures produce different results, our algorithm yields more homogeneous behavior between change points.

[2] The transforms and number of bins were chosen by hand to best capture important features of actual packet size and inter-packet interval distributions. In the future we hope to find techniques that allow us to avoid quantization.

[3] The essential property of this transformation is that it is a linear operator from the functional space of CDFs to $\mathbb{R}^n$. When converting vectors back into CDFs, they must be normalized them so that their last coordinate is unity.

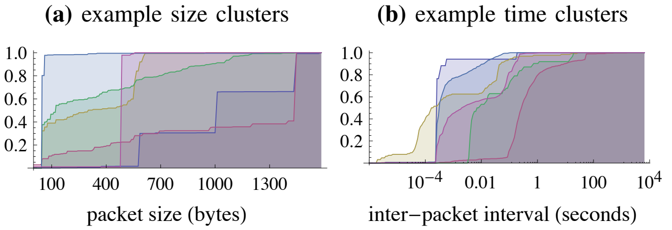

Figure 1: CDFs of example flowlet clusters for size and time.

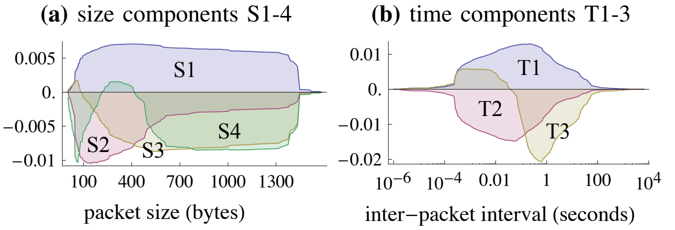

Figure 2: Normalized CDF residuals for principal components of size and time behavior. The average CDF across all clusters is subtracted from each component's CDF. The difference is normalized so that the area between the residual and the x-axis is unity.

PCA is that after transformation, the coordinates of the data are uncorrelated. However, they are not necessarily statistically independent. Essentially non-linear dependencies may remain between the PCA-transformed coordinates. See [15] for in-depth discussion of PCA and related techniques.

We apply PCA to our flow data in two different ways. First, we use PCA to find new bases for the collections of size and time clusters. This essentially gives us a small set vectors such that linear combinations of these vectors can be used to closely approximate all of the original cluster CDF-vectors. Only the initial b_s, b_t basis vectors required to explain 99% of the variation across the size and time clusters are retained. The new basis vectors, written with respect to the old coordinates form rotation matrices, $R_s \in \mathbb{R}^{c_s \times b_s}$ and $R_t \in \mathbb{R}^{c_t \times b_t}$, which can be used convert vectors of cluster coefficients to the new bases. The flow-cluster matrix, F, can be easily converted to the new coordinates by right-multiplication:

$$F' = FR_{st} \in \mathbb{R}^{f\times(b_s+b_t)}, \text{ where} \qquad (1)$$

$$R_{st} = \begin{pmatrix} R_s & 0 \\ 0 & R_t \end{pmatrix} \in \mathbb{R}^{(c_s+c_t)\times(b_s+b_t)}. \qquad (2)$$

Thus, each flow's behavior is concisely encoded as just $b_s + b_t$ coefficients of the new basis vectors.

This first stage of PCA extracts the common behavior across clusters into the matrix R_{st}, thereby allowing each flow's behavior to be expressed in terms of fewer cluster-like vectors. Common linear structure across all of the flows, however, may still exist. Before proceeding, we center each column by subtracting its mean: let $\mu_j = \frac{1}{f}\sum_{i=1}^{f} F'_{ij} \in \mathbb{R}^{b_s+b_t}$ and define $F''_{ij} = F'_{ij} - \mu_j$. We use PCA again to find a set of basis vectors for the rows of F'' such that only 1% of the variance in behavior across flows is lost. Let $a \in \mathbb{N}$ be the number of basis vectors required, and let $R_x \in \mathbb{R}^{(b_s+b_t)\times a}$ be the corresponding rotation matrix (its rows are the new basis vectors with respect to the old coordinates). We can then transform the set of flows one final time: $F''' = F''R_x \in \mathbb{R}^{f\times a}$. Each row of this matrix still corresponds to a single original flow, but the coordinates are now a linear transformation of the original $c_s + c_t$ cluster coefficients onto only a dimensions. The new coordinates are uncorrelated and explain the total variance of the flow behaviors in order of decreasing significance.

IV. Results & Discussion

To test our methodology, we use traces recorded in an infrastructured 802.11g wireless LAN with 18 access points, deployed at the 60th Internet Engineering Task Force meeting (IETF60), held in San Diego during August of 2004. The details of how the trace was captured and processed can be found in [2] and [3]; a collection of application-level flow traces are extracted from raw `tcpdump` trace files. These flow traces provide the raw data for our analysis. We use the flows from a 210-minute subset of the full traces. This subset contains 4,138 flows, but we restrict our analysis to the 902 flows with at least 10 packets, since statistical tests such as BWS lack adequate power for smaller samples than this.

Example size and time clusters, produced by the flow splitting and flowlet clustering algorithms, are shown in Figure 1. These examples represent only six of each cluster type, out of 518 size clusters and 552 time clusters, but they give a sense of "typical" clusters. Only six size principal components (PCs) and three time PCs explain 99% of the variation in their respective collections of cluster behaviors. The net effects of various size and time PCs on behavior distributions are illustrated in Figure 2. The first size component, S1, for example, increases the proportion of a broad range of mid-sized packets relative to very small and very large packets. As another more complex example, the third time component, T3, increases the proportion of smaller inter-packet intervals while drastically reducing the proportion of large ones.

Since there are six size and three time PCs after the first

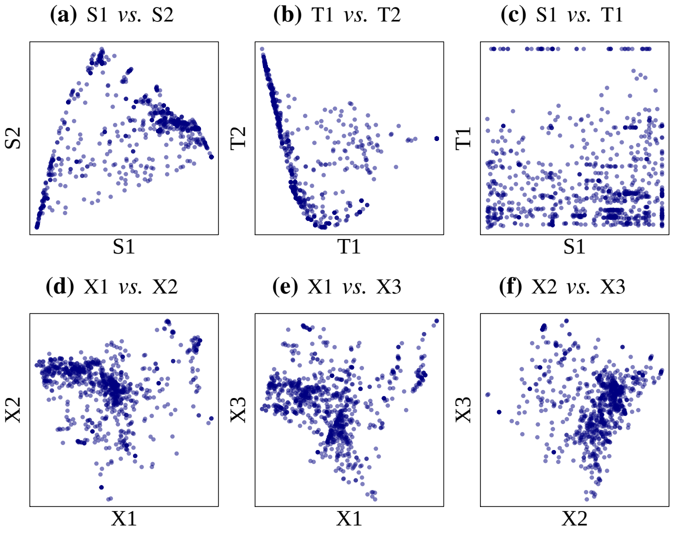

Figure 3: Scatter plots of flow behaviors projected onto various pairs of time (T1-2), size (S1-2), and joint time-size (X1-3) components.

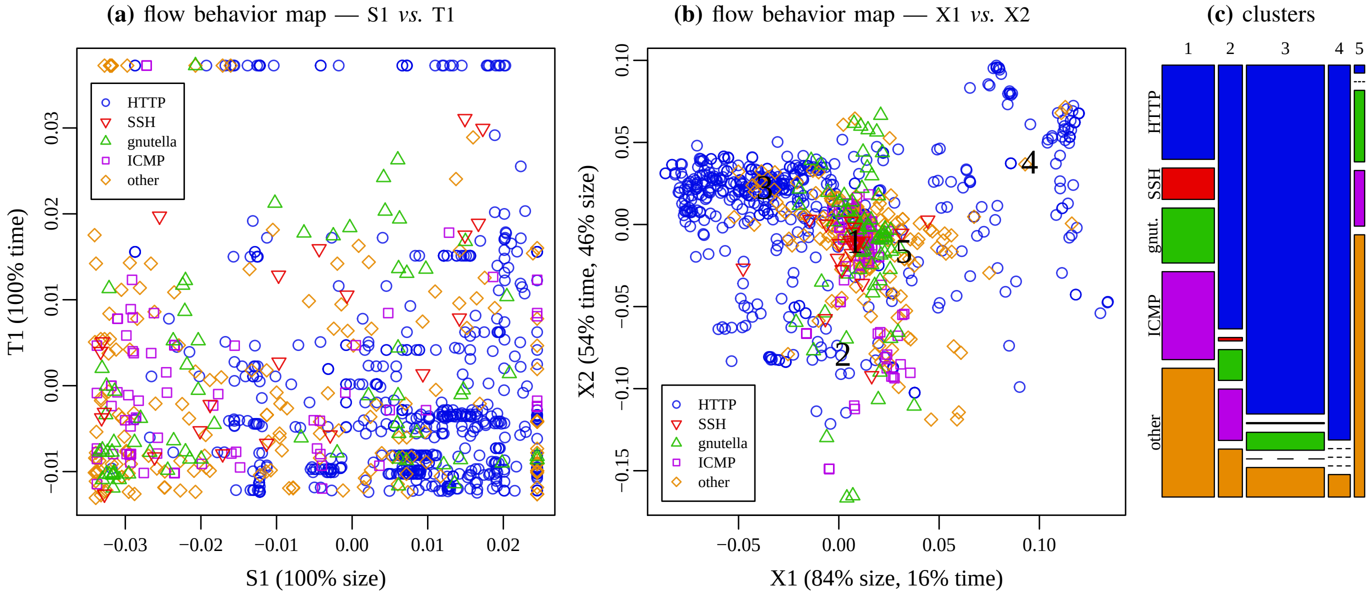

Figure 4: Flows separated by application type projected onto the planes defined by two different pairs of principal behavior components. Overlaid numbers in 4B show the projected centroids of flow-behavior clusters (not to be confused with flowlet clusters) found in joint PCA space using standard k-means clustering. The relative sizes of the clusters and their composition are shown in 4C as a mosaic plot.

stage of PCA, each flow can be represented in only nine dimensions. However, a great deal of structure remains in the interactions between the components, as seen in the scatterplots in Figures 3A-3C. The plots map various pairs of coefficients for all the flows in our data set. While the PCA produces linearly uncorrelated values, Figures 3A and 3B clearly show a high degree of non-linear interaction. In Figure 3C, on the other hand, the size and time behaviors appear mostly independent. The non-linear patterns in Figures 3A and 3B bear further attention. The curved portion of the convex hull of each pattern is approximately where vectors of the form $(0, \dots, 0, 1, \dots, 1)$ are mapped by the change of basis rotation. The other points can be written approximately as weighted sums of points on the hull in all PC dimensions (in two dimensions, it must be true by the definition of convexity).

After the second round of PCA, the nine separate size and time dimensions are mapped onto eight joint dimensions, each of which is partially time-like and partially size-like. These joint components can recreate 99% of the variance in the original time and size components. Moreover, they are uncorrelated, normalized, and each have zero mean. Figures 3D-3F show scatter plots of the first three joint components, X1, X2, and X3, against each other. The striking non-linear interactions that are so apparent in Figures 3A and 3B are no longer evident, although some degree of visible dependence remains. This is a significant improvement since we can approximate the distribution of the most significant joint components more readily with less dependence between the higher PCs.

As an application of our methodology, we now consider how different traffic types are mapped onto various traffic behavior components. The major types of traffic found in the IETF60 trace are HTTP, SSH, gnutella, and ICMP. Figure 4 shows two different behavior maps. Figure 4A uses S1 and T1 as coordinates, while Figure 4B uses X1 and X2. Each view has its advantages. The coordinates of the size-time map have intuitive interpretations, given by the behaviors associated with components S1 and T1. For example, flows in the lower-right portion of the plot have mid-sized packet sizes and inter-packet intervals that tend to be either very large or very small. This is consistent with the dominance of HTTP traffic in this region: the small packets are TCP setup and tear-down messages and GET requests, while the larger packets are HTTP responses.

The joint PCA map has no such immediate interpretation, since the components X1 and X2 are linear combination of many time and size components. On the other hand, in the joint map, natural clusters of flow types emerge. Moreover, standard multi-dimensional analysis techniques can be applied to the full PCA space. The overlaid numbers in Figure 4B mark the centroids of five clusters found using k-means clustering on the rows of F'''. Even though the clusters are derived based only on the time-space behavior flows, clear patterns with respect to traffic types emerge.

Once the matrices T, S, R_{st}, R_x and the vector μ have been derived from a collection of traffic traces, new flows can be quickly and easily mapped into the size-time or joint PCA spaces. First, quantize the empirical CDF of the flow's packet size and inter-packet interval values using the method described in Section III-D. Let $x_s, x_t \in \mathbb{R}^n$ be the quantized CDF vectors for size and time, respectively. Next, find coefficient vectors $\beta_s \in \mathbb{R}^{b_s}$ and $\beta_t \in \mathbb{R}^{b_t}$ that minimize the quantities

$$\|SR_s\beta_s - x_s\| \text{ and } \|TR_t\beta_t - x_t\| . \tag{3}$$

Here $\|\cdot\|$ denotes the standard Euclidean norm. This is a linear sum-of-squares minimization problem, which can be solved quickly and efficiently by any numerical linear algebra

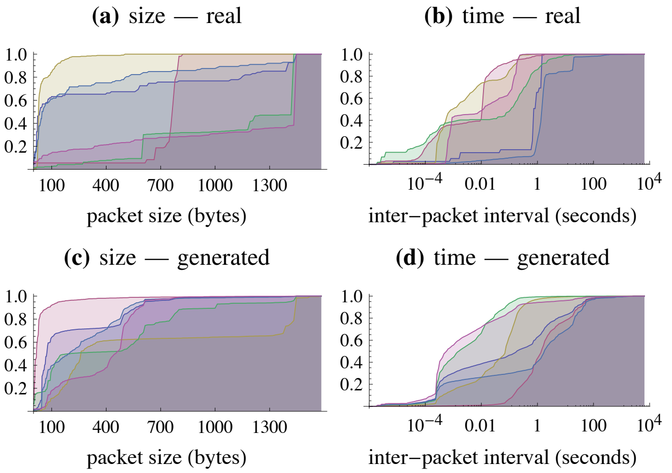

Figure 5: Random flow behaviors — real and generated.

system. Note that the matrices SR_s and TR_t can be precomputed. The vectors β_s, β_t already give us our size and time representations of the flow. Let $\beta = (\beta_s, \beta_t) \in \mathbb{R}^{b_s+b_t}$. To transform into joint PCA space, we simply subtract μ and right-multiply by R_x: $\alpha = (\beta - \mu)\, R_x$. The vector α is the joint PCA representation of the flow. On modern computers, these vector and matrix operations are extremely fast, allowing us to produce dynamic flow behavior maps in real time. This capability could be extremely useful for network monitoring or anomaly and attack detection.

Finally, we turn to the problem of generating realistic collections of flow behaviors. The coordinates of flows in joint PCA space can be very roughly approximated as a multivariate normal distribution, since the coordinates are uncorrelated and have zero means. This approximation is very rough, and not remotely statistically rigorous, but still, very useful—and possibly just good enough. Using this rough model, we can generate random multivariate normal deviates in joint PCA space and reverse transform them to get semi-realistic flow behaviors. Explicitly, if $\alpha^* \in \mathbb{R}^a$ is a randomly generated joint PCA vector, then we can transform it via matrix inversion:

$$(\gamma_s^*, \gamma_t^*) = \left(\alpha^* R_x^{-1} + \mu\right) R_{st}^{-1} \in \mathbb{R}^{c_s+c_t}. \qquad (4)$$

The vectors γ_s^*, γ_t^* are relative weights for size and time cluster atoms. Negative weights are artifacts of imperfect dependence modeling and should simply be ignored. The actual size and time CDFs are produced from γ_s^* and γ_t^* (after fixing up) via left-multiplication by the matrices S and T. Alternately, the actual empirical CDF for each cluster can be randomly sampled, and the weights used to chose randomly between the clusters. Figure 5 compares six randomly selected real flow behaviors with six random synthetic flows generated using this method. The resemblance is fairly good. While this method is somewhat rough, no other method currently exists for generating realistic full-network collections of flow behaviors based on real trace traffic. It remains to be seen if the realism of flows generated this way is sufficient by the standards defined in [2] and [3]. We intend to investigate precisely this question in our future research.

V. Conclusions & Future Work

This paper presents a fundamentally different way of viewing and understanding network trace data. The most radical concept entailed in this approach is that of finding common "atoms" of flow behavior and using these as the basis for representing and analyzing network-wide traffic patterns. The statistical and algorithmic methodology for finding these atoms of behavior is a significant contribution in itself, with application well beyond the field of traffic analysis. Once the atoms of behavior are extracted, it becomes natural to view the space of behaviors as a vectors space generated by these units. This representation puts a vast array of powerful linear algebraic tools at our disposal for understanding and analyzing network traffic. We have only begun to scratch the surface of the possible applications of this atomic theory of flow behavior.

References

[1] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multi-level traffic classification in the dark," in *Symposium on Communications Architectures and Protocols (SIGCOMM)*, Philadelphia, PA, USA, August 2005.

[2] S. Karpinski, E. Belding, and K. Almeroth, "Towards realistic models of wireless workload," in *3rd Workshop on Wireless Network Measurements (WinMee)*, Limassol, Cyprus, April 2007.

[3] ——, "Wireless traffic: The failure of CBR modeling," in *4th International Conference on Broadband Communications, Networks, and Systems (Broadnets)*, Raleigh, NC, USA, September 2007.

[4] M. Papadopouli, H. Shen, E. Raftopoulos, M. Ploumidis, and F. Hernández-Campos, "Short-term traffic forecasting in a campus-wide wireless network," in *Personal, Indoor and Mobile Radio Communications*, vol. 3, Berlin, Germany, September 2005, pp. 1446–1452.

[5] F. Hernández-Campos, M. Karaliopoulos, M. Papadopouli, and H. Shen, "Spatio-temporal modeling of traffic workload in a campus WLAN," in *2nd Annual International Workshop on Wireless Internet*, Boston, MA, USA, August 2006.

[6] M. Ploumidis, M. Papadopouli, and T. Karagiannis, "Multi-level application-based traffic characterization in a large-scale wireless network," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Helsinki, Finland, June 2007.

[7] M. Karaliopoulos, M. Papadopouli, E. Raftopoulos, and H. Shen, "On scalable measurement-driven modelling of traffic demand in large WLANs," in *IEEE Workshop on Local and Metropolitan Area Networks*, Princeton, NJ, June 2007.

[8] W. Baumgartner, P. Weiß, and H. Schindler, "A nonparametric test for the general two-sample problem," *Biometrics*, vol. 54, pp. 1129–1135, September 1998.

[9] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, April 1993.

[10] A. Inoue, "Testing for distributional change in time series," *Econometric Theory*, vol. 17, pp. 156–187, 2001.

[11] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *1st ACM SIGCOMM Workshop on Internet Measurement*, San Francisco, CA, USA, November 2001.

[12] M. Allen, J. Brevik, and R. Wolski, "Comparing network bandwidth time series," in *MetroGrid Workshop*, Lyon, France, October 2007.

[13] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[14] T. Ruefli and R. Wiggins, "Technical note: Longitudinal performance stratification—an iterative Kolmogorov-Smirnov approach," *Management Science*, vol. 46, no. 5, pp. 685–692, May 2000.

[15] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, "Proper orthogonal decomposition and its applications—Part I: Theory," *Journal of Sound and Vibration*, vol. 252, no. 3, pp. 527–544, May 2002.