Traffic Anomaly Detection at Fine Time Scales with Bayes Nets

Jeff Kline, Sangnam Nam, Paul Barford, David Plonka, and Amos Ron

***Abstract*—Traffic anomaly detection using high performance measurement systems offers the possibility of improving the speed of detection and enabling detection of important, short-lived anomalies. In this paper we investigate the problem of detecting anomalies using traffic measurements with fine-grained timestamps. We develop a new detection algorithm (called S3) that utilizes a Bayes Net to efficiently consider multiple input signals and to explicitly define what is considered "anomalous". The input signals considered by S3 are traffic volumes and correlations between ingress/egress packet and bit rates. These complementary signals enable identification of an expanded range of anomalies. Using a set of high precision traffic measurements collected at our campus border router over a 10 month period and an annotated anomaly log supplied by our network operators, we show that S3 is highly accurate, identifying 86% of the anomalies listed in the log. Compared with well known time series-based and wavelet-based detectors, this represents over a 20% improvement in accuracy. Investigation of events identified by S3 that did not appear in the operator log indicate many are, in fact, true positives. Deployment of S3 in an operational environment supports this by showing *zero false positives* during initial tests.**

I. Introduction

Whether malicious or unintentional, traffic anomalies are a fact of life in wide area networks. At a high level, the impact of anomalies is to reduce network performance and reliability, and as such, they are the bane of network operators. The standard process for addressing network anomalies is *detect - diagnose - remedy*. Therefore, improving any part of this process should improve network performance and reliability.

The general objective of our work is to improve the ability to *detect* network traffic anomalies in operational networks. While a wide variety of both ad hoc and automated methods for detecting anomalies are currently used, two of the most important requirements for detection are *accuracy* and *timeliness*. An accurate detection method raises an alert if and only if an anomalous event occurs in the network. Likewise, a timely detection method raises an alert soon after an anomalous event begins.

There are many challenges to accurate and timely anomaly detection. First, anomalies are difficult to define. While attacks, outages, flash crowds and misconfigurations are examples of anomalies writ large, specifics vary from network to network depending on operational policy. Second, non-anomalous traffic has complex characteristics. The inherent burstiness, and diverse and evolving composition of network traffic render some simple volume-based detection methods ineffective. Third, the ability to detect anomalies is intrinsically tied to the measurements that are available in a network. In general, these include SNMP MIB and flow-export data (such as Cisco's NetFlow) that are typically only collected at intervals of minutes which inherently limits the capability of any detection algorithm both in terms of accuracy and timeliness.

In this paper we address the problem of detecting anomalies accurately using time series traffic measurements collected using high performance monitors. We restrict ourselves initially to volume measurements because of their proven utility in prior work and the fact that they do not necessitate packet header or payload inspection. We argue that these measurements offer the possibility of detecting important short-lived anomalies, detecting anomalies in near real time, and exposing details of anomalies that could be used for classification. Our specific objectives are to develop an anomaly detection method that can consider multiple dimensions of high precision traffic measurements, which we hypothesize will lead to high detection rates, and to do this in a computationally tractable manner that can be used in operational environments.

The new anomaly detection methodology that we describe in this paper uses four time series with high precision timestamps (ingress/egress packet and bit rates). The first component of the detector uses a wavelet-based method to identify significant changes in volume of ingress and egress packet counts (what we refer to as *smoothness*). The second component of the detector is based on the premise that "normal" ingress and egress traffic exhibits a characteristic correlation structure. The central notion of this "normal" assumption is that the ingress and egress bit counts can be approximately recovered, in a time-stationary way, from the ingress and egress packet counts. We refer to this assumption as *traffic symmetry* which expands on the idea of *packet symmetry* that has been shown to be effective for DoS detection in prior studies [1], [2]. We then identify anomalous traffic by measuring the "distance" of the observed traffic from the normal one. We use certain parameters of a singular value decomposition of ingress and egress bit and packet counts as a means for measuring that distance. The use of two such *completely complementary* methods (smoothness and symmetry) enables our detector to capture a range of anomalous behavior beyond what either method in isolation could detect. The third component of the detector combines the first two to generate alerts using a Bayes Network, which is computationally efficient and enables what is "anomalous" to be defined explicitly and in a principled

J. Kline, S. Nam, D. Plonka, P. Barford and A. Ron are members of the Computer Science Department at the University of Wisconsin - Madison. E-mail: {kline,nam,plonka,pb,amos}@cs.wisc.edu

This work was supported in part by NSF grants CNS-0347252, CNS-0627102 and CNS-0646256. Any opinions, findings, conclusions, or other recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF.

fashion. Since this algorithm is based on traffic Symmetry, Singular value decomposition and Smoothness, we call it the *S3* detector.

To assess the capabilities of S3, we observed traffic over a period of 10 months on the commodity traffic link of our campus border router using an Endace DAG packet monitor [3]. We created the four input time series at one second granularity in near real time. A journal of traffic anomalies was painstakingly created over this period by our network operations group using coarse-grained SNMP and flow-export data. The journal contains details of 94 confirmed and diagnosed traffic anomalies. While this set of anomalies is assumed to be incomplete (due to the limitations of the coarse-grained data and operator attention), it nevertheless represents a standard for ground truth in false negative analysis of anomaly detectors.

Toward our objective of using the S3 detector in an operational context, we implemented it to test for anomalies over 15-minute windows at one minute steps (*i.e.,* fixing the minimum reaction time for detection at one minute). The fact that all components of the detector have low computational complexity enables the detector to function quite efficiently at this time scale. Our S3 detector identified 86% of the logged anomalies and 345 (average of 1.3/day) additional events. Manual inspection of many of the additional events reveals that nearly all are, in fact, *true positives* that were not visible to the operators who created the anomaly log with coarse-grained data. We believe that the rigor of the Bayes approach, its natural mechanism for specifying what is "anomalous", and its tunability offer a significant opportunity to both improve detection rate and lower false alert rate in operational deployments.

To further evaluate the S3 method, we implemented a standard Holt-Winters time series detector (HW) similar to [4] and a Deviation Score wavelet-based detector (DS) similar to [5], which was shown to perform very well in [6]. Both of these detectors identify "significant" fluctuations in volume of a single signal. After extensive manual tuning of HW and DS detectors for the fine time scale data, we were able to achieve a 65% detect rate with 443 additional events (average of 1.7/day) identified by HW, and a 63% detect rate with 588 additional events (average of 2.2/day) identified by DS. The 20% improvement by S3 over these prior methods (at similar, manageable additional alert rates) is thus significant and supports our hypothesis that considering multiple complementary characteristics of input signals leads to improved detection accuracy.

The final step in our investigation of S3's capabilities is an initial case study in which we asked a network operator to inspect all of the alerts generated by S3 over a one week period. During this time, S3 identified 8 anomaly episodes. The network operator investigated and diagnosed these and determined that *all* alerts were true positives. This result lends important additional support to the utility of our approach.

In summary, this paper makes the following contributions: *(i)* an initial examination of anomaly features that are exposed at fine time scales, *(ii)* introduction of a new anomaly detection method based on multiple complementary signals and a Bayes Network that is both accurate and timely, and offers opportunities for site specific tuning and continued refinement, and *(iii)* evaluation of the new method along with two standard methods for anomaly detection in traffic over a long duration with an operator supplied anomaly log.

II. Related Work

Network anomaly detection has been an active area of research for some time. Early efforts were focused on the practical problem of network fault detection [7], [8], and in using time series methods to detect traffic anomalies [9], [4]. Several studies have shown that entropy-based methods can be effective for anomaly detection [10], [11] including Xu *et al.* who use entropy to classify traffic in packet traces taken from an ISP backbone [12]. These information entropy methods rely on packet header information and thus require more processing per packet and more maintenance of state information than our method based on traffic rate inputs. (Our method does not preclude the use of packet content; it could provide additional time series input signals.) Bayesian Networks have been used to detect anomalies at the TCP connection level (*e.g.,* [13], [14]). Our work differs in that it employs Bayes Networks trained for normalcy using characteristics of aggregate traffic; we were unable to find prior examples of this application in the context of network traffic anomaly detection.

More recent work by Barford *et al.* explores the use of wavelets as the basis for detecting anomalies in NetFlow data [5]. The smoothing function used in our detector is similar to the mid frequency filter developed in that work. That study also used an annotated anomaly log as the basis for testing their detector. Lakina *et al.* pioneered the use of SNMP and NetFlow data sets from multiple sites and Principle Components Analysis (PCA) as the basis for *network-wide* anomaly detection [15], [16], [17]. Recent work by Ringberg *et al.* provides important insights on the difficulties in tuning network-wide PCA-based detectors in practice [18]. Two additional studies that describe promising methods for network-wide anomaly detection include [6], [19]. Our detection method could easily be adapted for use in a network-wide detection framework such as [6]. Finally, a recent reference to singular value decomposition in the context of traffic anomaly detection appeared in [20]. The basic formulation of the anomaly detection method described in that work-in-progress differs substantially from ours.

There is a large literature on the characteristics of network packet and flow traffic. Several more recent studies have investigated correlations between size, duration, rate and burstiness in traffic flows [21], [22]. There have also been many studies of network traffic based on packet traces collected with high performance monitors similar to those used in our study (*e.g.,* see [23]). However, we are not aware of any study that has systematically examined correlation structures between ingress and egress traffic on a link, which is an important component of our anomaly detector.

The idea of using packet symmetry for the purpose of denial of service attack detection was introduced by Mirkovic *et al.* in [1]. That work identifies malicious activity if the smoothed

ratio of packets sent to packets received exceeds a simple threshold on a per flow basis. Kreibich *et al.* expand the idea of packet symmetry by proposing that it be adopted as a principle of protocol design [2].

III. Data Collection

Our study is motivated in part by the simple fact that the process of anomaly detection is inherently limited both in accuracy and timeliness by the precision of the measurement system used to gather data. Since SNMP and flow-export measurements are typically only available at one to five minute intervals, events that take place on shorter time scales are unlikely to be visible in this data.[1]

We hypothesized that there are important anomalous traffic events that are discernible only with fine time scale data, and that this data exposes key features of events that could be useful for diagnosis and remedy. Furthermore, time series detection methods that are based on temporal aggregation are directly tied to the number of data points that are available for evaluation. For example, the wavelet-based deviation score method described in [5] is limited significantly by the fact that only 288 data points are available each day from data provided at five minute intervals. Measurements at a finer time scale should clearly improve timeliness, but should also have the potential to improve accuracy in aggregate-based detection methods.

A. Instrumentation

The fine time scale data used for this study was collected using a dedicated monitoring machine running Linux 2.4 with an Endace DAG4.3GE network monitoring card, dual Intel Xeon processors, and multiple SCSI disks. The monitoring card was connected to an optical tap on a link from one of our campus border routers to our primary Internet service provider. This link carries most of the campus' commodity Internet traffic. We used DAG driver software version 2.5.3 release 1 and a patched version of NeTraMet [24] software, version 5.1 beta 9 (NeTraMet51b9) to extract bit/packet time series. A patched version of RRDtool [25] version 1.2.12 was used to store the time series data.

NeTraMet was configured with a rule set that stored ingress and egress packet and bit rate values once every second. We experimented with different aggregates and found that one second increments resulted in a manageable volume of data with extremely rich features as will be shown below. The one second rate values from NeTraMet were continually downloaded from the DAG monitor via SNMP (on fifteen second intervals), post-processed and pushed into a round-robin-database (rrd) file. Once in the database, these rate "signals" could be accessed by our detectors.

B. Measurement Data

Our measurement infrastructure was used to collect data almost continuously between April 17, 2005 and January 22, 2006. The raw data was stored in an rrd file configured with a Round-Robin Archive (RRA) large enough to hold one year's worth of ingress and egress packet and bit rates: four floating point values per second. The average ingress and egress packet rates were 31K and 32K packets per second, respectively. The average ingress and egress bit rates were 139M and 189M bits per second, respectively. The only significant discontinuity in the data is a measure outage period of two weeks in May and June 2005, during which our measurements are missing.

Similar to [5], a journal of actual traffic anomalies that occurred in our network was meticulously maintained during the measurement data collection period. The entries in this journal were the "major" anomalies identified during this period and in each case were carefully diagnosed by the network operations staff. The sources used to identify the anomalies were five-minute granularity SNMP and flow-export data visualized by MRTG [26] and FlowScan [27], respectively. Although this limits an operator's ability to to distinguish very short-lived events (thus the log cannot be used for false positive analysis), this set of events is *ground truth* for evaluating the performance of our anomaly detection method and comparing it with other detectors. This stands in contrast to evaluation methods used in prior work such as relying on other anomaly detectors to identify candidate events (*e.g.,* [15]) or relying on injected artificial anomalies into traces (*e.g.,* [28]).

Each entry in the anomaly log notes the event's date and time, a short description of the anomaly's root cause, and sometimes an end time. This enabled us to group the events into the following categories:

- **Abuse**: Typically a Denial-of-Service event, usually flood-based. For instance: an outbound flood of UDP packets from a campus host that has had its security compromised and is being remotely controlled by a malicious party.
- **Flash**: A flash crowd event. For instance: hundreds of clients outside our campus receiving a live video stream of a sporting event sourced from a server on campus.
- **Measurement**: An anomaly due to legitimate traffic or measurement system failures. For example: a campus host participating in TCP bulk data transfer with a host at another campus as part of a research project.
- **Network**: A network failure event or temporary misconfiguration resulting in a connectivity problem or outage. For instance: a scheduled code upgrade on our service provider's router.
- **Unknown**: An anomaly for which evidence was found, but the root cause was not identified.

Engineers operating the campus network identified, researched, and tagged events resulting in a log of 94 confirmed anomalies suitable for evaluating our detection method. [2] This journal served as a road map to points of interest in the traffic rate database, and enabled us to build a query engine to select from this data.

C. Sample Anomalies

The time series data collected in this study offers an unusual perspective even for experienced network operators since most

[1] For a specific example, see the anomaly in Figure 1.

[2] The distribution and counts of these anomalies are shown in Section V.

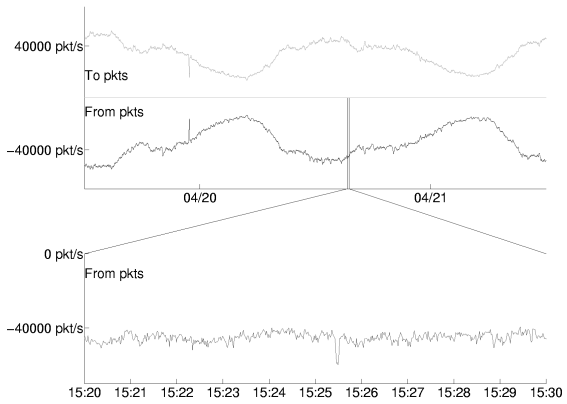

Fig. 1. Example of a rapid probe of an entire class B network. This anomaly is invisible in 5 minute aggregate data. 48 hour perspective at 5 minute granularity (top), 10 minute perspective at one second granularity. (bottom).

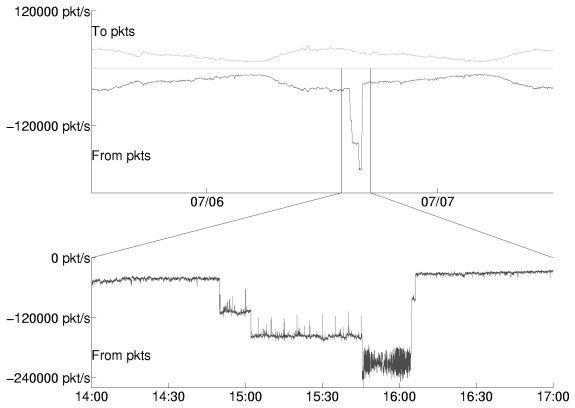

Fig. 2. Example of a progressively more intense UDP flood anomaly from three hosts: 48 hour perspective at 5 minute granularity (top), 2 hour perspective at one second granularity (bottom).

networking equipment is not capable of exposing traffic rates in so timely or granular a fashion. We present three examples of events that highlight the level of detail afforded by the fine time scale data and motivate our use of traffic rate measurements as a basis for automated anomaly detection. While each of these sample anomalies exhibit a dramatic change from normal in packet rates and some ingress versus egress packet asymmetry, many other unique characteristics become visible at the finer time scale. These additional features may well serve as a basis for improving detection, or anomaly classification in future work.

Figure 1 shows the inbound and outbound packet rates when a host in the outside world sent a single 46 byte packet to the TCP port 512 (the "exec" service) of each IP address in one of our campus' class B networks. Presumably this was to elicit a response in preparation to attempt to compromise campus hosts' security and is therefore likely malicious abuse of the network. The top graph shows the inbound and outbound rates as five minute averages (typical for SNMP or flow-export data) over 48 hours; the probe traffic is not discernible. The lower graph shows the rate in one second averages over 10 minutes; the spike representing that abusive probe traffic is prominent. This event exemplifies the ability of fine time scale data to expose short-lived events of interest.

Figure 2 shows the inbound and outbound packet rates when three hosts in the outside world sent a flood of UDP packets destined for ports 21 and 80 of one campus host. The top graph shows the rates as five minute averages over 48 hours; an increasingly large spike in inbound traffic is clearly evident. The lower graph shows the rate in one second averages over 2 hours; a series of irregularly-sized steps is present, suggesting that each of the source hosts began flooding at a different time, that all three flooded simultaneously, and that the second host was the last to cease flooding. This kind of coordinated behavior is suggestive of botnet activity (*e.g.,* [29], albeit at a small scale) and the detail provided by fine time scale data could be useful in the diagnosis process.

Figure 3 shows the inbound and outbound packet rates

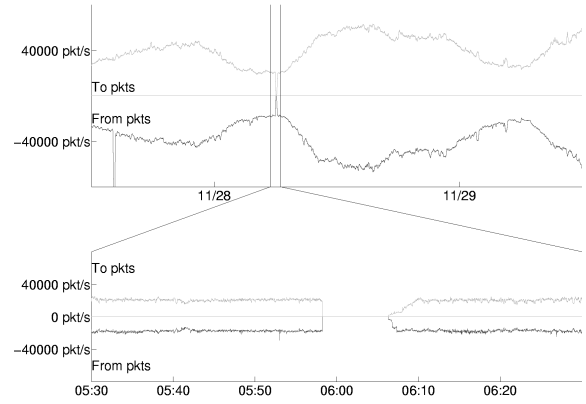

Fig. 3. Example of a network anomaly due to routine maintenance : 48 hour perspective at 5 minute granularity (top), 20 minute perspective at one second granularity (bottom).

during a scheduled network maintenance outage. The top graph shows the rates as five minute averages over 48 hours; a drop to zero is clearly evident in the mid-point of the graph. (The earlier inbound spike is an unrelated anomaly.) The lower graph shows the rate in one second averages over 20 minutes; a gradual return of the traffic is evident. Unlike the prior examples, this event is strongly symmetric with respect to packet rates. This highlights a challenge for a detector based only on identifying asymmetric changes in traffic.

IV. Building a Detector

Our process for building an anomaly detector has two steps. The first is identifying the set of signals whose behavior can be correlated with anomalous traffic. This step can be decomposed into (i) identifying a time series that can be extracted from the high performance monitors, and (ii) applying a transformation to this time series that enhances the signal to noise ratio. The second step in the process is to develop a detector that is applied to the signal set to identify anomalies by distinguishing "abnormal" from "normal".

The model for our anomaly detector is based on considering multiple complementary input signals as a means for improv-

ing accuracy. Our first observation is that analysis of individual time series (*e.g.*, ingress packet rate) can be quite effective for identifying anomalous spikes in traffic streams that arise for a variety of reasons such as large scale attacks and network outages. Our second observation is that there are inherent correlations between normal ingress and egress traffic from a network due to the network's population of consumers (which pull data into the network) and providers (which push data out of the network), bidirectional applications, and reliable transport protocols. We refer to this relationship as *traffic symmetry* which could include aggregate packet, byte, or flow rates or more detailed breakdowns of traffic afforded by full packet traces. Traffic symmetry is a strong and persistent characteristic in our data. We posit that important classes of anomalous events such as changes in the dominant application or even subtle adjustments in the application mix may not be evident in volume measures but will be reflected in detectable changes in traffic symmetry. While traffic symmetry may not be as useful for exposing anomalies on *e.g.*, core routing paths, we expect that segments that form gateways between networks will be good candidates for taking advantage of this important property.

A. From Measurements to Signals

The measurements we considered in this work – ingress/egress bit and packet count time series – are straightforward to produce by passive measurement and it is well known that changes in these streams can correlate with important types of anomalies. While many other time series from packet monitors may be useful for detecting traffic anomalies, we leave the analysis of these to future work. It is important to note that our detector framework described below can easily be expanded to include additional time series. Our task was to develop transformation methods for basic data streams that lead to accurate identification of anomalies.

1) First component: smoothness estimator: This component examines each of the ingress/egress packet count time series, and actually measures the *lack of smoothness* (*i.e.*, related to variance) in the data. This is a standard procedure: first, the time series is decomposed into a hierarchy of *detail coefficients* (via a wavelet decomposition), and organized according to time scale. Then the rate of decay of these coefficients as we move from the largest time scale to the smaller scales is estimated. The higher the decay, the higher the smoothness. The working assumption is that certain anomalies increase the volatility of the traffic volume, thereby reducing the smoothness parameter. We are particularly interested in periods when the smoothness decreases simultaneously for the incoming as well as outgoing packet counts. Symmetric decrease of smoothness may be indicative of irregular behavior that may *not* radically distort symmetries in the data volumes (*e.g.*, Figure 3), hence these events may not be identified by the second component of the detector, described below.

The calculation of the smoothness estimator takes place over a specified time interval. While we experimented with several possible values, a 15 minute interval was selected for our implementation. Within a given interval, we extract the ingress and egress packet count at 1 second sample rates over the interval $[T, T+899]$. While we could get the smoothness readings themselves at 1 second granularity (by shifting the 15-minute window 1 second at a time), we observed that readings at 1-minute steps captures accurately the variability of smoothness, whenever it occurs.

Each time series is then decomposed into a hierarchy of detail coefficients using a wavelet system. (The Daubechies 6-tap wavelet system [30] was chosen for our analysis.) Next, we compute the norms of the detail coefficients by taking the square root of the squared sum of the coefficients and obtain a sequence of norms corresponding to the wavelet hierarchy. The smoothness estimation of each time series is then computed by taking a weighted sum of these norms with larger weights applied to the norms of the smaller time scale coefficients. This is a discrete version of the Sobolev norm. The result is a nonnegative real number that can be arbitrarily large.

Specifically, for fixed wavelet $\psi := \psi^{Daub,6-tap}$, we used the following as our definition of the Sobelov norm:

$$\|f\|_{W^s} := \left(\sum_{k,m} \left(1 + 2^{-2sk}\right) \left|\langle f, \psi_{k,m}\rangle\right|^2 \right)^{1/2}$$

where $\psi_{k,m} : x \mapsto 2^{-k/2}\psi(2^{-k}x - m)$. For our analysis, we set $s := 1/2$.

2) Second component: correlation estimator: In contrast to the first component which is obtained by "averaging local behavior", this component is genuinely global. The first step in correlation analysis is to choose a time interval for the analysis. Like the smoothness estimator, we experimented with different intervals, and found that a 15 minute period works quite well. We normalize the volume readings in a given window and obtain four vectors: *IP* (Inbound/ingress Packet rate), *OP* (Outbound/egress Packet rate), *IB* (Inbound/ingress Bit rate), *OB* (Outbound/egress Bit rate). Each vector has 900 entries (900 seconds = 15 minutes). For example, $IP(i)$ records the ratio between the number of incoming packets that arrived during the ith second and total number of incoming packets, but normalized with the ℓ_2-norm. That is, if $v(i)$ records the number of packets that arrived at the ith second then:

$$\mathrm{IP(i)} = \frac{v(i)}{\sqrt{\sum_{j=1}^{900} v(j)^2}}.$$

A thorough assessment of these values for our data shows that the above four vectors are strongly correlated. This is due to the fact that the large majority of the measured data corresponds to normal bidirectional application traffic. It is reasonable to assume that relations similar to those we observed will be found in data collected at other locations in the Internet (gateway links in particular).

A prominent characteristic of our data is that the four normalized vectors are well-approximated by a 2-dimensional plane. This characteristic is observable over a wide range of time scales, times of day and volumes in traffic. While this 2-dimensional approximability phenomenon may seem surprising at first, a closer examination of prevailing network characteristics show this phenomenon to be the case under the following realistic assumptions. The first assumption is

that packet count and byte count are dominated, in non-anomalous periods, by large TCP-based flows (so called elephant flows [22]). The second assumption is that the outgoing TCP traffic during a short period of time (*e.g.,* 15 minutes) is *homogeneous*, *i.e.,* tends to exhibit an almost constant ratio between the number of outgoing data packets and the number of incoming acknowledgment packets. Let us denote that near-constant ratio by R_1. A similar assumption is made for the incoming TCP traffic, but quite likely with a different ratio R_2 (*i.e.,* we allow the data flow in one direction to have different characteristics as compared to the data flow in the opposite direction). Finally, we assume that the average size of a packet (along the measured period, *i.e.,* typically 15 minutes) depends on the direction (ingress/egress), and whether the packet contains data or is a ACK with no payload, but is largely independent of other factors. These combined assumptions yield that, with the vectors of incoming packets V, and the vectors of outgoing packets W, we observe on the ingress link a vector similar to

$$V + R_2 \cdot W,$$

and on the egress link

$$R_1 \cdot V + W.$$

Moreover, the corresponding byte-count vectors are

$$A \cdot V + B \cdot R_2 \cdot W,$$

and

$$C \cdot R_1 \cdot V + D \cdot W,$$

for suitable constants A, B, C, D. All these four vectors lie in $\mathrm{span}(V, W)$. Thus, under the realistic assumptions stated above, the four collected time series, during non-anomalous periods, lie in a two dimensional plane. The normalization process that we apply to each vector does not change this important property. At the same time, one has good reason to expect that anomalous traffic (*e.g.,* DoS attacks) will be very different in its characteristics, and hence will distort the aforementioned dependencies among the four collected time series.

To test the symmetry assumption on four vectors *IP*, *OP*, *IB*, *OB*, we used first the singular value decomposition (SVD) to compute a suitable approximating 2D plane. Let e_1 and e_2 be the two singular eigenvectors that span this plane. Our analysis is based on the eight inner products between each of the vectors *IP*, *OP*, *IB*, *OB* on the one hand, and the two eigenvectors e_1, e_2 on the other hand.

We now describe our hypothesis concerning the output of the above analysis under "ideally normal" traffic conditions. Our first assumption of ideal normal traffic is that the eigenspace plane is spanned by *IP* and *OP* (and not only by their projections), and that the eigenvectors e_1, e_2 are chosen based on *IP* and *OP* alone (without involving *IB* and *OB*). This assumption is reasonable since the common applications create high correlation between *IP* and *OP*, far higher than any correlation that might be expected between *IB* and *OB*. This assumption was also validated extensively in our own data. Under those ideal conditions we must then have:

$$IP = a_1 e_1 + a_2 e_2, \quad OP = a_1 e_1 - a_2 e_2,$$

for some scalars a_1, a_2 that satisfy $a_1^2 + a_2^2 = 1$. In reality, we only have:

$$P(IP) = a_{11} e_1 + a_{12} e_2, \quad P(OP) = a_{21} e_1 - a_{22} e_2,$$

with P the orthogonal projection on the 2D eigenspace, and with:

$$D_1 := a_{11} a_{21} + a_{12} a_{22} < 1.$$

Our first assessment of "distortion from normalcy" is based on the value of $1 - D_1$. A large value here corresponds to a higher degree of violation of our normalcy assumption.

Our second hypothesis of normal behavior is that the vectors *IB* and *OB* are uncorrelated, and they both completely lie in the eigenspace plane. In reality, it is only true that:

$$P(IB) = b_{11} e_1 + b_{12} e_2, \quad P(OB) = b_{21} e_1 + b_{22} e_2,$$

with b_{ij} the inner products between the vectors *IB*, *OB* and the eigenvectors e_1, e_2. Had our hypothesis been true, the Gram matrix (b_{ij}) would have been unitary, and hence its determinant (in absolute value) must have been equal to 1. Our second measure of normalcy is based on this determinant:

$$D_2 := |b_{11} b_{22} - b_{12} b_{21}|$$

The difference $1 - D_2$ provides us with a second measurement of "distortion from normalcy".

We noted earlier that our two estimators are based on complementary methodologies: one looks for local distortions along each time series, while the other looks for global inconsistencies among the entire collection of time series. As a matter of fact, the complementarity of the two methods is even more fundamental than the above description may indicate. The smoothness of each time series is governed primarily by the initiation and termination of short-lived flows, as well as by flows that behave erratically. Most of the data volume does not belong to these flows, and is filtered out by the wavelet tool. Large values from the smoothness estimator is an indication of unusual *increase* in erratic behavior, (or in the initiation/termination rate) of short lived flows. Our assumption is that an "anomaly" creates spontaneous erratic behavior, and at the same time distorts the model of elephant's dominance. Thus, we actually estimate two properties that are somewhat similar, but by completely disjoint analytical tools.

B. Bayes Net Detection

We implemented a detection method that uses the smoothness and symmetry estimates as input to a Bayes Network [31]. The Bayes Net method offers a systematic means for establishing differences between normal and anomalous behavior with multiple input signals. Bayesian Networks are a general-purpose tool used to infer causal relationships between multiple random variables. More specifically, they are graphical models which use directed acyclic graphs to compactly represent joint probability distributions. Each random variable represents a node on a graph; directed edges indicate that a dependence exists between a pair of nodes.

Our data is a six-dimensional time series: two correlation values and four smoothness values for each point in time. To

infer the relationships between these values and their baseline behavior, a training set of observations is required.

Special care is necessary when establishing baseline behavior. Traffic observed at noon on a typical weekday cannot be used to characterize traffic measured during early morning hours on that same weekday. Similarly, times during the weekend may not be fairly used as a baseline for traffic during weekdays. Finally, vacation days also exhibit behavior that does not mimic typical weekdays. For a university setting, this means distinguishing between summer and semester weekdays.

Although we collected data continuously, we only included the semester weekday data when training and testing our Bayes Net detector. Each weekday was divided into sixteen disjoint 90-minute periods. This partitioning attempts to limit the effects of standard diurnal cycles in network traffic.

We used the the MATLAB Bayes Net Toolbox for this portion of S3. To create a Bayes network of a size that would be suitable for an operational deployment, we encoded the smoothness and correlation signals as integer values. The smoothness signal was reduced to a boolean value while the correlation signals were encoded as integers ranging from 1 to 4. Specifically, for each 90 minute period in a day, the 5th and 95th percentiles of the smoothness estimators were found and saved. Values below the 5th percentile or above the 95th percentile are considered "in the tails" and therefore candidates as anomalies. These values were then assigned a boolean identifier indicating whether or not they occurred in the tails. The correlation estimators were organized into 1 of the following 4 categories: [0, 0.7), [0.7 0.8), [0.8, 0.92) and [0.92, 1].

Weekdays of the entire 10 month data were thusly partitioned and encoded. The encoded data were used to estimate the six dimensional (empirical) joint distribution for each of the 90 minute periods. "Normal" behavior is therefore assumed to be the dominant characteristic in the encoded data. Conversely, "anomalous" behavior is assumed to have occurred infrequently, although we do not distinguish this in the training data so it is also encoded in the joint distribution. The key point here is that *the joint distribution captures and quantifies normalcy*.

To identify an anomaly, we proceed as follows. Given a current observation, we compare the frequency of occurrence of similar events in the past. Whenever such events occurred sufficiently infrequently (this is reflected as a tiny value in the joint distribution), the alarm is raised. To set this alarm threshold, a collection of 20 candidate events from the operators log of varying magnitude, duration and type were selected. A threshold of 1e-4 was sufficient to identify all of the candidate events while remaining minimally sensitive.

Selection of the values 0.7, 0.8, and 0.92 used for encoding the correlation values was based on the observation that the candidate anomalies we wished to identify assumed values within each of these ranges. The 5th and 95th percentiles used for encoding the smoothness data were fixed without respect to the canonical anomalies.

The time interval/window size and the number of wavelet levels are closely related to each other. This pair of parameters serves to influence the S3 detector's sensitivity to short-lived events. Shorter windows increase the sensitivity. The number of available wavelet levels is limited by the size of the time window. We usually set the number of wavelet levels to be approximately $\log_2(WindowSize)$. The size of the window shift determines the granularity of the analysis. A shift that is smaller than the size of the time window provides redundancy in the analyzed data. For example, a window of 15 minutes coupled with a shift of 1 minute means that every point in time is examined 15 times.

A list of the parameters used in the S3 Bayes Net detector evaluation is given in Table I. We plan to investigate automated methods for tuning the detector in future work.

TABLE I
S3 BAYES NET DETECTOR PARAMETER LIST AND SETTINGS USED IN THIS STUDY.

Parameter Description	Parameter Value
Window Size	900 seconds
Wavelet Levels	8 levels
Window Shift	60 seconds
Day Partition Size	90 minutes
Smoothness Partitions	0.05 and 0.95 percentiles
Correlation Partitions	0.7, 0.8 and 0.92
Detector Threshold	1e-4

TABLE II
EVALUATION RESULTS: COUNT OF CANDIDATE ANOMALIES DETECTED BY THE HOLT-WINTERS, DEVIATION SCORE, AND S3 METHODS.

Anomaly Type	Candidate Count	HW Detected	DS Detected	S3 Bayes Detected
Abuse	76	54 (71%)	54 (71%)	71 (93%)
Flash	2	1 (50%)	0 (0%)	0 (0%)
Measurement	8	4 (50%)	4 (50%)	4 (50%)
Network	4	2 (50%)	1 (25%)	4 (100%)
Unknown	4	0 (0%)	0 (0%)	2 (50%)
Total	94	61 (65%)	59 (63%)	81 (86%)

V. DETECTOR EVALUATION

We evaluated S3's performance and compare its capabilities relative to two other anomaly detection techniques. To do this we had to first build and then tune these other detectors to operate on fine time scale data. Tuning is a fact of life for all statistical anomaly detectors. Our objective in tuning the detectors used in this study was to identify as many of the ground truth anomalies from the operator's journal as possible without an explosion in the number of additional events detected. Our aim in terms of the latter was to keep the average number of additional events detected on the order of 1 to 2 per day over the 266 day data set. We emphasize that it is unreasonable to call all of the additional events detected by any of the methods "false" positives since through manual analysis, we found that in many cases the events are true short-lived anomalies that were simply not visible in the operators' coarse-grained time series plots of SNMP or flow-export data.

A. Methodology

To evaluate each anomaly detector's performance, we matched the anomalies they reported with anomalies in the

operator's log. To do this, we first selected a subset of the logged anomalies that occurred within the time range of our measurement data. (Some occurred during the two-week measurement outage period.) Then, for each candidate anomaly, we determined an *evaluation window* beginning one minute prior to the time the event was logged and ending according to either the event's duration (if the operators' log supplied this information) or three minutes after it began. That is, for short-lived anomalies, matches were evaluated within a four minute window. This approach tolerates small time stamp errors in the operator's log. It also tolerates some detector latency, within a window of time ranging from a minimum of a few minutes to a maximum of the event's duration as determined by a network operator.

B. S3 Detector Performance

The right-most column of Table II summarizes the detection performance of our S3 method indicating 86% success in identifying the logged anomalies. The application of each detector was made in one minute steps, thus an anomaly might be detected within one minute of its inception. While the S3 detector performed very well, it did not identify several symmetric anomalies, such as an exchange of HTTP traffic at a very high rate between a client and server. In each case, S3 signaled a distortion that we believe will be recognized by future versions of the detector.

We argued in Section IV that the complementary smoothness and correlation signals in S3 provide an important benefit over either method in isolation. To highlight the value of classifying based on both these inputs, we note the following. The smoothness detector with the 0.82 threshold identified 984 total events in our traffic, and 81 of the 94 logged anomalies. Likewise, the symmetry detector with the 9e9 threshold identified 507 total events in our traffic and 77 of the logged anomalies.

Without complete ground truth, we cannot say anything absolutely definitive about the 345 additional events reported by the detector that did not appear in the operator's log. If it were the case that the additional alarms are all false, the respective average rates over the 266 day period are 0.67/day and 1.3/day which are still likely to be acceptable to network operators.

We visually inspected all of the additional events and made the following observations. Many were found to be duplicates of another event due to windowing issues. This can easily be addressed in future work. Others were large inbound or outbound spikes in packet rate, and resembled ABUSE events such as flood-based attacks. Still others were instances in which traffic dropped to zero, and were likely to be short-lived NETWORK outage events. Another sort of anomaly reported by S3 but not found in our operator's log were prolonged episodes of packet rates fluctuating more than usual. The root cause of this phenomenon is unknown, but it is clearly evident in one instance as an asymmetric event (with an additional 10k inbound packets per second on average) and in another event as a symmetric fluctuation in the packet rates.

C. Holt-Winters Time Series-based Detector Performance

To gain perspective on the capabilities of S3, we tested an alternative detector which has been shown to perform well on coarser time scale data: the aberrant behavior detection tool available in RRDtool. This tool is based on Holt-Winters time series forecasting and identifies anomalous behavior based on a prediction and a confidence band of varying width, each of which is influenced by the past values, taking into account both seasonal (*e.g.* daily) variations and recent behavior. (Note, that similar EWMA methods are used to establish ground truth for other traffic anomaly studies *e.g.*, [15]) Guided by results from prior studies, we used this technique on the ingress and egress packet rate time series.

Our implementation was based on RRDtool version 1.2.12, but is essentially the same what was used in [4]. Various modifications to the model parameters were required to handle our traffic rate data at one second intervals. When constructing the RRDtool database, the "HWPREDICT", "SEASONAL", "DEVPREDICT", and "DEVSEASONAL" Round-Robin Archives (RRAs) were configured by trial and error with alpha = 0.000385, beta = 0.000012, gamma = 0.000385, delta = 5[3], and a seasonal period of 86400, which is one day expressed in seconds. The "FAILURES" RRA was configured with a threshold of 45 and a window of 60. This means that a minimum of 45 violations (observed values outside the confidence bounds) within a window of 60 seconds was considered a high-confidence "anomaly". Theoretically, this method then has the ability to detect an anomaly within one minute of its inception. A Holt-Winters reported anomaly was considered a match to a logged event if the RRDtool "FAILURES" RRA had a non-zero value any time within the given event's evaluation window.

The Holt-Winters (HW) detector found 65% of the candidate anomalies from the log as shown in Table II. Overall, the HW detector reported a total of 270 anomalies in the egress packet rates and 234 anomalies in the ingress packet rates for a total of 443 additional alarms (average of 1.7/day). Some of these are likely to be false positives, but a number of these are the same suspected ABUSE or NETWORK events that were identified by S3.

D. Wavelet-based Detector Performance

We also implemented and evaluated the performance of a Deviation Score (DS) detector. This detector was based on wavelet decomposition of packet count time series as described in [5], but configured to detect anomalies at finer time scales. This general method was also shown to perform very well in [6].

After trying several different parameters for computing the deviation scores, we noted that improving the detection rate without increasing the total hit count was quite difficult. This behavior is similar in some ways to the smoothness indicator

[3]We chose a delta value of 5, greater than the values prescribed by [32] and [4], each of which suggest values between 2 and 3, to reduce the number of false reports that occurred frequently due to the bursty packet rates observed when using an interval of one second interval instead of several minutes, as used in prior work.

of the S3 method. This is somewhat expected because both calculations of the smoothness and the deviation scores involve decomposing the underlying signal into different frequency components and capturing "strength" of each component at each time point. However, it is important to point out that the two have the following differences. First, in the deviation scores several frequency levels from the wavelet decomposition are combined and normalized while the smoothness calculation in S3 does not involve such operations. Second, different time window sizes are used for the calculations of the "strengths" (local variances) in different components in the deviation scores while there are no such parameters in the S3 smoothness.

For the purpose of comparison, we computed the deviation scores as follows. Packet streams were decomposed into 15 frequency levels using the Daubechies 6-tap wavelet system [30]. Three sets of 5 consecutive frequency levels were combined into the hi-frequency, mid-frequency, and low-frequency components. Time windows of sizes 2^5 seconds, 2^{10} seconds, and 2^{15} seconds were used to compute the local variances of hi-frequency, mid-frequency, low-frequency components, respectively. Equal weights were used to compute the final deviation scores.

Tuning of DS requires simultaneous selection of multiple parameters. Unfortunately, a comprehensive sweep through the DS parameter space is impractical. The parameters presented in this discussion were found by analyzing small samples of "normal" data as well as a handful of anomalies we expected all detectors to identify. We used a threshold of 5e-5 on the deviation scores to detect anomalies. Note this is a huge change from the original DS threshold of 2.3 reported in [5] and is due to the difference in the data granularity. The DS detector identified 63% of the candidate anomalies as shown in Table II. Overall, the DS detector reported 234 anomalies in egress packet rates and 413 anomalies in ingress packet rates for a total of 588 additional events identified (average of 2.2/day). Manual inspection of these indicated that many are false positives.

E. Results Discussion

In comparing the S3 detector results with those of the HW and DS detectors, we find that S3 performed better overall providing an 86% detect rate of the anomalies listed in the operator's journal versus 65% for the next best detector with similar additional alarm rates. Of the journaled anomalies, most were ABUSE anomalies; S3 detected 93% of these important events, as opposed to 71% for HW and DS. We argue that this provides strong support for our hypothesis that combining smoothness and correlation signals in fine time scale data leads to meaningful improvement in overall detection rates.

One reason for HW method's poorer detection performance appears to be that, as configured, it quickly adjusted or displaced its confidence band when a spike occurred and the increase was sustained for minutes. This made the detector less sensitive than the S3 detector to subsequent anomalies which occurred just minutes after another. This would appear to be a fundamental limitation of this time series-based method when applied to network traffic at fine time scales.

The number of additional events detected were approximately the same for S3 and HW. As mentioned above, this was a function of our threshold selection for each algorithm where we attempted minimize false negatives and maintain additional alerts to an average of approximately 1 to 2 per day. However, DS reported significantly more additional events when tuned to approximately the same detect rate as HW, raising the suspicion that it is more likely to report false positives. We attribute this performance problem to DS only having one output signal that performs similar to the S3 smoothness estimator alone. Compounding DS' false positive reports, it produced many spurious alarms in weeks that contained many real spike-based anomalies with similar magnitude. We attribute this to be an artifact of the weekly periodic normalization it employs. Conversely, we also attribute some of the DS detectors false negatives to normalization since a large spikes dwarf other anomalies that occur within the week, reducing their deviation score. This accounted for a substantial portion of the DS false negatives. These shortcomings suggest that anomaly detectors based solely on smoothness measures or that rely on periodic normalization may be limited in their effectiveness.

F. Online S3 Bayes Performance

Since the evaluation reported above leaves open the question of whether the reported events were false positives or merely oversights in the historical operator's log, we performed an initial near real time evaluation of S3. Over five days, January 22 through January 26 2007, we ran the S3 Bayes detector (trained with a subset of the 10 month data set and tuned as described above), and used its reports to direct an operator's attention to anomalous periods of traffic for investigation and verification.

To perform this evaluation, we used the same measurement system described in Section III. It was configured to record 72-byte frame headers for all traffic and simultaneously record the DAG ports' frame and byte rates at one-second intervals, more directly than the NeTraMet method used earlier. These values were stored into a round-robin-database file. During the test period, the measured link carried an average of 280 megabits per second and about 55,000 frames per second in each direction. We ran a self-contained analysis script to generate the smoothness and symmetry values and then apply the Bayes detector. This script was run continuously over small batches of data, and could trivially be applied as frequently as deemed useful by a network operator.

During the Monday through Friday test period, the detector reported eight events in non-overlapping 15-minute windows. (Interestingly, this is the same daily event detection rate that the S3 Bayes reported for the 10 month data set.) Subsequent investigation by the network operator found that six of these eight episodes contained ABUSE anomalies: inbound or outbound short-lived packet floods or network probes. The remaining two episodes contained otherwise unnoticed NETWORK anomalies in which the traffic rate dropped to zero, for some seconds, in one or both directions on the

measured link. This evaluation provides further support for the accuracy (by avoiding false positives) of our method and its feasibility for continuous use in an operational environment.

VI. Conclusions and Future Work

In this paper we investigate the problem of traffic anomaly detection using traffic measurements at fine time scales. We develop a new detection algorithm for this data that considers both changes in volume and changes in correlations of ingress and egress traffic on a link, and raises alerts using a Bayes Net. We evaluate the capability of our S3 detector on an extensive time series of traffic rates taken at our campus border router using a high performance passive measurement system. Our dataset also includes a log of actual anomalies that was maintained by our network operations group throughout the course of our study. While we experimented with longer intervals, we configured our detector to generate alerts in one minute intervals that should be quite sufficient for operational use. In terms of accuracy, the S3 detector was shown to provide over a 20% improvement on prior methods demonstrating the value of our new approach.

Our detector also identified a set of events that took place on short time scales but did not appear in the set of operator logged anomalies. The configuration of S3 used in this study reported fewer than two of these events per day over the course of our study. While we cannot say with certainty whether these are true anomalies, on closer inspection, most of them bear the hallmarks of familiar anomalies such as malicious scans and other types of abuse. A one week case study using a prototype on-line version of S3 supports this position showing *zero false positive alarms*. Regardless of their causes, it is clear that these events that take place on short time scales have rich characteristics that warrant future investigation. More generally, our results demonstrate that anomaly detection with fine grained data provides a novel and valuable perspective for network operators. While 20% improvement in detection accuracy is substantial itself, we consider this a starting point for further gains that should be possible when combining fine-grained measurements with our probabilistic reasoning framework.

While we've demonstrated that our S3 detector works well with time series inputs synthesized from packet and bit rates, we envision that its Bayes Net performance could be enhanced by utilizing additional inputs derived from flow counts or packet header and payload data. We plan to investigate these possibilities in future work.

References

[1] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," in *Proceedings of 10th IEEE International Conference on Network Protocols*, November 2002.

[2] C. Kreitich, A. Warfield, J. Crowcroft ans S. Hand, and I. Pratt, "Using Packet Symmetry to Curtail Malicious Traffic," in *Proceedings of ACM/USENIX Hotnets '05*, College Park, MD, November 2005.

[3] Endace Measurement Systems, "DAG Network Monitoring Interface Card," http://www.endace.com, 2005.

[4] J. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring," in *Proceedings of the USENIX Fourteenth System Administration Conference LISA XIV*, New Orleans, LA, December 2000.

[5] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[6] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network Anomography," in *Proceedings of ACM SIGCOMM Internet Measurement Conference '05*, Brekeley, CA, October 2005.

[7] C. Hood and C. Ji, "Proactive Network Fault Detection," in *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[8] I. Katzela and M. Schwartz, "Schemes for Fault Identification in Communications Networks," *IEEE/ACM Transactions on Networking*, vol. 3(6), pp. 753–764, December 1995.

[9] F. Feather, D. Siewiorek, and R. Maxion, "Fault Detection in an Ethernet Network Using Anomaly Signature Matching," in *Proceedings of ACM SIGCOMM '93*, San Francisco, CA, September 2000.

[10] W. Lee and D. Xiang, "Information-theoretic Measures for Anomaly Detection," in *Proceedings of IEEE Symposium on Security and Privacy*, Brekeley, CA, May 2001.

[11] Y. Gu, A. McCallum, and D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," in *Proceedings of ACM SIGCOMM Internet Measurement Conference '05*, Brekeley, CA, October 2005.

[12] K. Xu, Z. Zhang, and S. Bhattacharya, "Profiling Internet Backbone Traffic: Behavior Models and Applications," in *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.

[13] Alfonso Valdes and Keith Skinner, "Adaptive, Model-Based Monitoring for Cyber Attack Detection," in *RAID '00: Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, London, UK, 2000, pp. 80–92.

[14] S. Staniford, J. Hoagland, and J. McAlerney, "Practical Automated Detection of Stealthy Portscans," *IOS Press*, vol. 10, pp. 105–136, 2002.

[15] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-wide Traffic Anomalies," in *Proceedings of ACM SIGCOMM '04*, Portland, OR, August 2004.

[16] A. Lakhina, M. Crovella, and C. Diot, "Characterization of Network-wide Anomalies in Traffic Flows," in *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, Taormina, Italy, October 2004.

[17] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," in *Proceedings of ACM SIGCOMM '05*, Philadelphia, PA, August 2005.

[18] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection," in *Proceedings of ACM SIGMETRICS*, San Diego, CA, June 2007.

[19] A. Soule, K. Salamatian, and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection," in *Proceedings of ACM SIGCOMM Internet Measurement Conference '05*, Brekeley, CA, October 2005.

[20] J. Terrell, K. Jeffay, F. Smith, L. Zhang, Z. Zhu, H. Shen, and A. Nobel, "Multivariate SVD Analyses for Network Anomaly Detection," in *ACM SIGCOMM Poster Session*, Philadelphia, PA, August 2005.

[21] K. Lan and J. Heidemann, "On the Feasibillity of Utilizing Correlations between User Populations for Traffic Inference," in *Proceedings of IEEE Internation Conference on Local Computer Networks*, Sydney, Australia, November 2005.

[22] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the Characteristics and Origins of Internet Flow Rates," in *Proceedings of ACM SIGCOMM '02*, Pittsburgh, PA, August 2002.

[23] Sprint Labs, "IPMON Project," http://research.sprintlabs.com, 2005.

[24] Nevil Brownlee, "NeTraMet beta versions," ftp://ftp.auckland.ac.nz/pub/iawg/NeTraMet/beta-versions/, 2005.

[25] Tobi Oetiker, "RRDTool," http://oss.oetiker.ch/rrdtool/, 2005.

[26] Tobi Oetiker, "MRTG," http://oss.oetiker.ch/mrtg/, 2005.

[27] D. Plonka, "FlowScan: A Network Traffic Flow Reporting and Visualization Tool," in *Proceedings of the USENIX Fourteenth System Administration Conference (LISA XIV)*, New Orleans, LA, December 2000.

[28] S. Kim, A. Reddy, and M. Vannucci, "Detecting Traffic Anomalies Through Aggregate Analysis of Packet Header Data," in *Proceedings of Networking*, Athens, Greece, May 2004.

[29] V. Yegneswaran, P. Barford, and V. Paxson, "Using Honeynets for Internet Situational Awareness," in *Proceedings of ACM/USENIX Workshop on Hot Topics in Networks (Hotnets IV)*, College Park, MD, November 2005.

[30] I. Daubechies, *Ten Lectures on Wavelets*, CBMS NSF Reg. Conf. Series in Applied Math. SIAM, 1992.

[31] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Upper Saddle River, NJ, 1995.

[32] A. Ward, P. Glynn, and K. Richardson, "Internet Service Performance Failure Detection," *ACM SIGMETRICS Performance Evaluation Review*, August 1998.