ASTUTE: Detecting a Different Class of Traffic Anomalies

Fernando Silveira[†⋆]   Christophe Diot[†]   Nina Taft[◇]   Ramesh Govindan[‡]

[†]Technicolor   [⋆]UPMC Paris Universitas
[◇]Intel Labs Berkeley   [‡]University of Southern California

ABSTRACT

When many flows are multiplexed on a non-saturated link, their volume changes over short timescales tend to cancel each other out, making the average change across flows close to zero. This equilibrium property holds if the flows are nearly independent, and it is violated by traffic changes caused by several, potentially small, correlated flows. Many traffic anomalies (both malicious and benign) fit this description. Based on this observation, we exploit equilibrium to design a computationally simple detection method for correlated anomalous flows. We compare our new method to two well known techniques on three network links. We manually classify the anomalies detected by the three methods, and discover that our method uncovers a different class of anomalies than previous techniques do.

Categories and Subject Descriptors: C.2.3 [Computer-Communication Networks]: Network Operations

General Terms: Experimentation, Measurement.

Keywords: Anomaly Detection, Statistical Test.

1. INTRODUCTION

Uncovering anomalies in large ISPs and enterprise networks is challenging because of the wide variety of such anomalies. Anomalies can come from activity with malicious intentions (e.g., scanning, DDoS, prefix hijacking), or from misconfigurations and failures of network components (e.g., link failures, routing problems, outages in measurement equipment), or even legitimate events such as unusually large file transfers or flash crowds.

A number of techniques have been proposed [2, 12, 14, 26, 29] in order to identify some of these anomalies by analyzing network traffic. They all seek to expose anomalies by detecting deviations from some underlying model of normal traffic. Usually, this model has to be learned from days or weeks of anomaly-free traffic traces, which is a practical

problem since the training data is never guaranteed to be clean and training should be performed periodically.

In this paper, we introduce a new approach to anomaly detection that does not require training a model from historical data. Rather, we use a relatively simple, but surprisingly effective, statistical test for inferring strong correlations among flows on a single link. This test is based on a mathematical model of a type of equilibrium which we call ASTUTE (A Short-Timescale Uncorrelated-Traffic Equilibrium, Section 2). Based on ASTUTE, we propose an anomaly detection technique (Section 3) that detects strongly correlated flow changes, i.e., events where several flows simultaneously increase or decrease their volume, even when these flows do not share common 5-tuple features such as IP addresses, ports, and protocol number. Many types of anomalies (e.g., scanning and DDoS attacks, link outages, routing shifts) exhibit this type of behavior. We show that such anomalies violate ASTUTE. Our detector has a single threshold parameter that directly controls its false positive rate (under certain statistical assumptions).

We compare our technique to two well known detectors: Kalman filter [26] and Wavelet [2]. We review both detectors in Section 4. We identify and classify nearly 600 anomalies found by ASTUTE, Kalman, and Wavelet in three traffic data sets (links from two research ISPs and one corporate network). Section 5 describes the traces and methodology used in our evaluation.

In Section 6, we study the characteristics of the anomalies found by each method. ASTUTE is more effective than Kalman and Wavelet at uncovering anomalies that involve a large numbers of flows (such as scanning attacks and link flaps) particularly when the aggregate packet volume in these flows is small. In fact, ASTUTE finds anomalies that involve one or two orders of magnitude fewer packets than anomalies found by Kalman and Wavelet. For anomalies with many correlated flows, ASTUTE achieves high detection rates with very low false positive probabilities. We also show that ASTUTE's sensitivity to low-volume anomalies comes with a trade-off; ASTUTE is oblivious to anomalies that involve a few large flows, which Kalman and Wavelet can easily spot. Finally, we perform controlled simulations via anomaly injection to evaluate how ASTUTE and Kalman trade-off true detections and false alarms.

ASTUTE is fundamentally different from other detectors because it does not need to learn a model of normal traffic; it merely infers strongly-correlated flows. Our results show that almost all of the flows inferred by ASTUTE to be strongly correlated in links carrying commercial, research,

and enterprise traffic are indeed anomalies. This is consistent with findings in prior work [1, 10] which have shown that most dependencies across flows observed on real links are normally very weak. Because of this different approach, ASTUTE has three unique advantages:

- Not requiring a training phase makes ASTUTE computationally simple and immune to data-poisoning.

- Since ASTUTE is specialized in a class of anomalies (strongly correlated flows), it is more accurate at flagging these anomalies than other detectors.

- Once an alarm is flagged, ASTUTE provides information about the anomaly that facilitates classification.

2. AN EQUILIBRIUM MODEL

We introduce a model for normal traffic behavior, called ASTUTE (A Short-Timescale Uncorrelated-Traffic Equilibrium). As the name suggests, ASTUTE's assumptions hold for aggregate traffic on a link under certain conditions: over short timescales (on the order of minutes), when a large number of independent flows traverse a non-saturated link.

2.1 Model Definition

A traffic *flow* is a set of packets that share the same values for a given set of traffic features (e.g., source and destination IP addresses, source and destination ports, and protocol number). To study the evolution of a flow, time is usually divided into fixed sized intervals called *bins*. The *volume* of a flow f during bin i, denoted by $x_{f,i}$, is the number of packets or bytes in the flow during the corresponding bin.

In the ASTUTE model, flows crossing a link of interest are generated by a *discrete-time marked point process* [7], where the mark process determines both the flow's duration and its volume per time bin. Thus, each traffic flow f is uniquely defined by the following random variables:

- s_f, the time bin where the flow arrives into the link;

- d_f, the number of bins where the flow is active;

- $\vec{x}_f = (x_{f,s_f}, \ldots, x_{f,s_f+d_f-1})$, a vector with the flow's volume for each time bin where it is active.

While our model allows any distributions in the arrival and mark processes (e.g., arrivals can be Poisson or not, flow sizes can be heavy tailed or not), we make the following two assumptions:

(A1) *Flow independence* - a flow's properties (s_f, d_f, and $\vec{x}_f$) are independent of other flows' properties.

There are two well-known ways through which flow independence can be violated. First, some flows can be grouped into sessions; e.g., after a client downloads a web page from a server, it may open connections to other servers to download objects contained in the page. Second, flows can be correlated during congestion episodes, since they share the same queues in routers. This can happen if a link is saturated, since some flows need to reduce their throughput so that other flows can increase theirs. Previous works [1, 10] have shown that, despite these two common reasons for correlation, the dependencies across flows observed in traces from real links are normally very weak. One of the reasons for this is that most backbone links are under-utilized, as they are over-provisioned by design.

(A2) *Stationarity* - the distributions of the flow arrival process and the mark process do not change over time.

Stationarity is heavily dependent on the timescale in which we observe flows, i.e., the size of time bins. Even though traffic exhibits strong non-stationary at long timescales (e.g., daily and weekly cycles, long-term trends), several works have shown that, at short timescales (i.e., less than an hour), traffic is well modeled by stationary processes [4, 21].

It is important to understand that ASTUTE does not contradict the observation that network traffic is self-similar [16]. One way in which self-similarity manifests itself is in slowly decaying correlations in measurements of *total* traffic volume across *time*. ASTUTE is agnostic to temporal correlations and only focuses on correlations across *flows* within a single time bin. Indeed, self-similar traffic can be explained by the superposition of a large number of independent flows [28], albeit with a specific characteristic (heavy tailed on-off times).

2.2 Consequences of the ASTUTE Model

Consider a pair of consecutive bins, i and $i + 1$. Let $\mathcal{F}$ be the set of flows that are active in i or $i + 1$. For $f \in \mathcal{F}$, let $\delta_{f,i} = x_{f,i+1} - x_{f,i}$ be the volume change of f from i to $i + 1$. If the flow starts at bin $i + 1$ (or finishes at bin i), we consider that $x_{f,i}$ (resp. $x_{f,i+1}$) is zero in the definition of $\delta_{f,i}$. Finally, let Δ_i be the set of $\delta_{f,i}$'s for each $f \in \mathcal{F}$. The following theorem summarizes the main consequences of our model, and is the foundation of our anomaly detector.

THEOREM 1 (CONSEQUENCES OF ASTUTE).
When both (A1) and (A2) hold, the variables in Δ_i are zero mean i.i.d. random variables. In order words, for arbitrarily chosen flows f and g in $\mathcal{F}$:

(a) $\delta_{f,i}$ *has zero mean;*

(b) *if $f \neq g$, then $\delta_{f,i}$ is independent from $\delta_{g,i}$;*

(c) $\delta_{f,i}$ *and $\delta_{g,i}$ have the same distribution.*

PROOF. To prove item (a) for an arbitrary flow $f \in \mathcal{F}$, condition on its duration $d_f = d$ and volume $\vec{x}_f = \vec{x} = (x_{s_f}, \ldots, x_{s_f+d-1})$. For fixed d and $\vec{x}$, the flow's arrival time s_f can range from $i - d + 1$ to $i + 1$. Given the stationarity of the flow arrival rate in (A2), f is equally likely to have arrived in any of the bins in this range, i.e., s_f is uniformly distributed between $i - d + 1$ and $i + 1$. We can then express the mean of $\delta_{f,i}$ by conditioning on all the possible values of s_f, each with probability $\frac{1}{d+1}$:

$$\sum_{s_f=i-d+1}^{i+1} \frac{\delta_{f,i}}{d+1} = \sum_{i=s_f-1}^{s_f+d-1} \frac{x_{i+1} - x_i}{d+1}$$
$$= \frac{x_{s_f+d} - x_{s_f-1}}{d+1} = 0. \qquad (1)$$

In the last step above, we use the fact that both x_{s_f+d} and x_{s_f-1} are zero since f starts at bin s_f and ends at s_f+d-1.

Item (b), on the other hand, is a direct consequence of assumption (A1). Namely, since the flow volumes $\vec{x}_f$ and $\vec{x}_g$ are independent for two different flows f and g, so must be their volume changes $\delta_{f,i}$ and $\delta_{g,i}$ between bins i and $i + 1$.

Finally, item (c) relies on (A2) and the above observation that the conditional distribution of $\delta_{f,i}$, given d_f and $\vec{x}_f$,

depends *only* on d_f and $\vec{x}_f$. Because of this observation, if the distributions of d_f and $\vec{x}_f$ do not depend on f, the marginal distribution of $\delta_{f,i}$ is also independent of f. $\square$

In the proof of item (c) above, note that we can have a single distribution of flow properties even if there are multiple "classes" of flows, where a class represents flows generated by a given application (e.g., HTTP, VoIP) or using a given transport protocol (e.g., TCP, UDP). Suppose there are a set of classes $\mathcal{C}$, and let the distributions of flow sizes and volumes depend on a flow's class. For each $c \in \mathcal{C}$, let α_c be the probability that a flow in the total traffic belongs to class c, and according to the stationarity assumption $(A2)$, let α_c be constant across time bins. The distributions of d_f and $\vec{x}_f$ across all flows can be considered as a mixture distribution [8], with weights α_c and mixture components given by the distributions of the individual classes.

3. ASTUTE-BASED ANOMALY DETECTION

In this section we describe an anomaly detector based on the consequences of the ASTUTE model (Theorem 1). We also provide empirical evidence that the assumptions in our detector are observed in real traffic.

3.1 Detection Method Description

To detect strongly correlated flows, we design an anomaly detector whose null hypothesis [15] is the set of consequences in Theorem 1. Namely, we test with high confidence whether the volume changes of flows are i.i.d. samples of a zero-mean distribution. For this, we simply compute the confidence interval for the average volume changes across flows, and check if that confidence interval includes zero. If this condition does not hold for a given time bin, we mark that time bin as anomalous. We consider only traffic *on non-saturated links, and using short-timescale bins.* The rest of this section formalizes this intuition.

Consider F flows that are active at bin i, with volume changes given by $\delta_{f,i}$. Let $\hat{\delta}_i$ be the sample mean and $\hat{\sigma}_i$ the sample standard deviation [8] of the volume changes, i.e.:

$$\hat{\delta}_i = \sum_{f=1}^{F} \frac{\delta_{f,i}}{F} \quad \therefore \quad \hat{\sigma}_i = \left[\sum_{f=1}^{F} \frac{(\delta_{f,i} - \hat{\delta}_i)^2}{F-1} \right]^{\frac{1}{2}}. \quad (2)$$

If Theorem 1 holds, then for large F, $\hat{\delta}_i$ has a $(1-p)$-confidence interval given by the central limit theorem [8]:

$$I_{\hat{\delta}_i} = [\hat{\delta}_i - K(p)\hat{\sigma}_i/\sqrt{F}, \ \hat{\delta}_i + K(p)\hat{\sigma}_i/\sqrt{F}], \quad (3)$$

where $K(p)$ is the percentile $1-p/2$ of the standard Gaussian distribution. We say that a set of flows satisfies ASTUTE if $I_{\hat{\delta}_i}$ contains zero. Otherwise, we say that there is an *ASTUTE anomaly* at time bin i.

Clearly, the efficacy of the algorithm depends upon the choice of $K(p)$. As we increase the confidence level $1-p$, we also increase the size of the confidence interval $I_{\hat{\delta}_i}$. For a given set of flows, it is clear from (3) that the size of the interval is characterized by the value of $K(p)$. The smallest value of $K(p)$ such that the interval contains zero is:

$$K' = \frac{\hat{\delta}_i}{\hat{\sigma}_i}\sqrt{F}. \quad (4)$$

We call K' the *ASTUTE assessment value* (AAV) of a time bin. Note that ASTUTE is violated if and only if $|K'|$

is larger than $K(p)$. This equivalence holds because, for a large set of flows that satisfy ASTUTE, the AAV distribution is close to a standard Gaussian distribution. Later in this section we perform simulations to determine the minimum number of flows needed to obtain this approximation.

When ASTUTE is violated there are two possibilities. First, the confidence interval $I_{\hat{\delta}_i}$ is supposed to contain zero only for a fraction $1-p$ of the time bins. Thus, in a fraction p of the time bins, we should expect ASTUTE to be violated by normal traffic. This is the *false positive rate* of our anomaly detection method and it can be reduced by increasing the detection threshold $K(p)$.

The second possibility is that some set of flows violates our model's assumptions of flow independence and stationarity (Section 2.1). Later in this section we show that stationarity holds for time bins smaller than 15 minutes. Thus at short timescales (i.e., between 1 and 5 minutes), an AS-TUTE anomaly *must be triggered by a violation of the flow independence assumption.* For instance, if many flows increase (or decrease) their volumes at the same time, then these flows are no longer independent of each other.

Several types of events can trigger ASTUTE anomalies. Some forms of attacks like DDoS, port scans, network scans generate several correlated flows. Operational problems like route flapping, link outages, and transient congestion on upstream links can also cause flows to correlate.

However, not all volume changes will result in ASTUTE violations. A single high-volume flow does not violate the independence assumption and will not cause an ASTUTE violation. Even though a single large flow makes the average volume change $\hat{\delta}_i$ deviate from zero, it also increases the standard deviation $\hat{\sigma}_i$ at the same rate, thereby keeping the AAV low. More generally, a small number of correlated flows does not trigger an ASTUTE anomaly. ASTUTE focuses on strong correlations, i.e., those involving simultaneous changes in several flows. The strength of the correlations that we can flag as anomalous is determined by the single parameter in our detector, the threshold $K(p)$. We explore this issue in greater detail in Section 5.2.1.

Our method is valid when the link under consideration is not saturated, i.e., even with anomalous traffic, the total link utilization is less than 100%. If the monitored link is fully saturated for two consecutive time bins, the average volume change (and thus the AAV) should be zero, in which case ASTUTE does not flag an alarm. However, anomalies do not necessarily saturate all links everywhere in the network. For example, a collection of anomalous flows may be constrained by some upstream link, before reaching the link where ASTUTE is deployed. We verify that this condition holds in all the time bins that we evaluate.

Our ASTUTE-based anomaly detection method (henceforth, simply called ASTUTE) can be summarized as:

Initialization:

> Given a target false positive rate p, determine the detection threshold $K(p)$ as discussed above.

For each pair of consecutive time bins, do:

1. For each flow f, measure its volume change between the two time bins, $\delta_{f,i}$.

2. Compute K', the AAV defined in Equation (4).

3. If $|K'|$ is larger than $K(p)$, flag an anomaly.

We track the AAVs for every pair of consecutive bins, in six flow aggregation levels: *5-tuples*, *source IPs*, *destination IPs*, *host pairs* (i.e., source and destination IPs together), *source ports*, and *destination ports*. We restrict our discussion to these six flow aggregations simply because they have worked well in our experiments; our methodology is general enough to allow other combinations of 5-tuple features, or even other types of header fields (e.g., MAC addresses). We consider a bin as anomalous if any of the flow aggregation levels triggers an ASTUTE anomaly. Tracking ASTUTE at different aggregation levels is useful for two reasons. First, it provides additional reliability to our detector, as some anomalies are easier to find in specific aggregation levels. Second, the fact that some anomalies are *not* visible at certain aggregation levels provides information that help us to find the set of correlated anomalous flows. We discuss both of these issues in more detail later in the paper.

3.2 Timescales with Stationary Behavior

To make sure that ASTUTE anomalies are violations of the flow independence assumption, we need to validate the stationarity assumption. Intuitively, at large timescales, stationarity is violated by daily patterns of link usage. To pinpoint the timescales in which stationarity holds, we run our anomaly detection method in a trace from the GEANT2 network (described later in Section 5.1) for different bin sizes. We then measure the probability that ASTUTE triggers an anomaly at each given time of the day, averaged over a month. Figure 1 shows this metric for a detection threshold equal to 6. We see that for 5-minute time bins, the probability of detecting an anomaly is uniform throughout the day, indicating it is not sensitive to time-of-day effects. However, for bin sizes larger than 15 minutes, there is a high chance of flagging anomalies when the number of users ramps up in the morning, or drops down in the evening. We have observed the same qualitative result for other traces and different values of the detection threshold. Therefore, in the rest of the paper, we use time bins of 5 minutes.
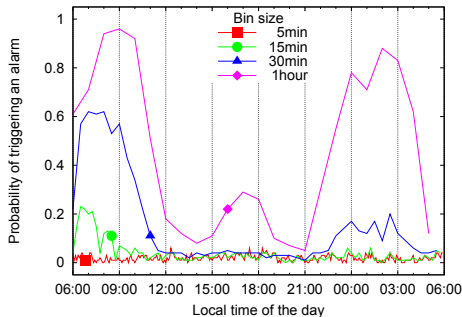

Figure 1: Non-stationarity violates ASTUTE for bins longer than 15 minutes.

3.3 Validating the Gaussianity of AAVs

Our detector relies on the fact that the AAVs of normal time bins follow a standard Gaussian. This important since it allows us to directly relate ASTUTE's false positive rate to its detection threshold $K(p)$. In this section, we validate this basic result by studying the impact of two characteristics of real traffic measurements: (1) aggressive packet sampling rates; and (2) the skew in flow size distributions.

3.3.1 Packet Sampling

Due to processing and storage overhead, traffic measurements in highly aggregated links often employ random packet sampling. Typical traces are sampled at rates as low as 1% or even 0.1%. Although we described ASTUTE in terms of non-sampled traffic data, our results are still valid under random packet sampling. We show this using a publicly available 48-hour long packet trace from a 100 Mbps link between Japan and the USA[1].

We bin 5-tuple flows into 5-minute intervals and compute the AAVs for each bin in the trace. Figure 2 shows QQ-plots comparing the distribution of AAVs, for different sampling rates, to the standard Gaussian quantiles. The main challenge in analyzing these plots is that we can never be sure that a trace contains only normal time bins. However, since anomalies tend to increase the AAV in absolute value, they should impact only the tails of the AAV distribution, and small AAVs should be closer to Gaussian. The QQ-plot for the non-sampled trace shows that the AAVs are well approximated by the Gaussian distribution in the range between -2 and 2. Outside this range, the AAV distribution deviates from Gaussian, indicating that the trace contains anomalies that violate ASTUTE. Note that as we sample packets, the AAVs become closer to Gaussian in the whole range, i.e., both head and tails of the distribution. In summary, the plots show that sampling does *not* violate ASTUTE, i.e., it does not induce false positives. On the other hand, sampling can make some low-volume anomalies disappear [18]. We have performed the same analysis on other traces and obtained similar results.
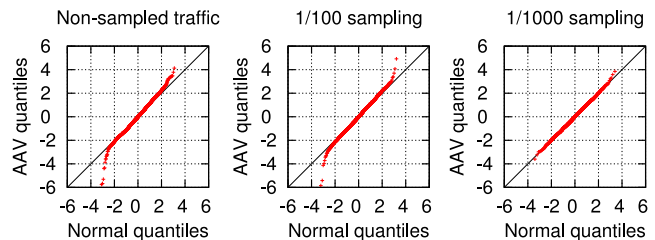

Figure 2: QQ-plots for empirical AAVs.

3.3.2 Flow Size Distribution

Since the Gaussianity of the AAVs is a consequence of the CLT, it only holds for flow size distributions with finite variance [8]. Although previous works have observed that total flow sizes are well-modeled by highly skewed distributions with infinite variance [5], our result depends only on the flow volume within finite time bins. This restriction imposes a natural limit on the maximum flow size, dictated by the bin size and the link's capacity. Because of this, the distribution of flow sizes within a time bin has finite variance and the CLT convergence has to occur given a large enough number of flows. We perform simulations using synthetic flow size distributions to visualize this convergence and the flow sizes from a real traffic trace to show that this convergence is achieved in practice.

Given a flow size distribution, we generate an i.i.d. sample of F flows in a pair of time bins and we compute the corresponding AAV. We repeat this 1,000 times, and compute

---

[1]MAWI archive - `http://mawi.wide.ad.jp/mawi/`

the distribution of the resulting AAVs. We then use a Filliben hypothesis test [9] to check if the AAV distribution is close to Gaussian. The statistic used by Filliben's test is the correlation between the AAV samples and the corresponding quantiles of the standard Gaussian. If this statistic is below a critical value (derived from simulations [9]) we can reject the hypothesis that the AAVs are Gaussian.

Figure 3 shows the test statistic (on the y-axis) for different flow size distributions (the different curves) and different values of F (on the x-axis). If the flow sizes have a non-skewed exponential distribution, the CLT convergence is extremely fast, as the correlation goes beyond the critical value even for ten flows. To observe the case of a highly skewed distribution, we simulate flow sizes using a Pareto distribution with infinite variance. The corresponding curve shows that the test statistic does not reach the critical value, even for tens of thousands of flows. However, if we bound the maximum flow size in the Pareto distribution to a hundred packets, then convergence to Gaussian is achieved once we have around a hundred flows per bin. Finally, we also perform simulations using the empirical flow size distribution from the GEANT2 trace (described later in Section 5.1). The corresponding curve goes above the critical value with at least a few dozens of flows per bin. In all of our traces, over 95% of the time bins contain more than 100 flows. The typical number of flows is much higher than that, i.e., on the order of thousands or tens of thousands.
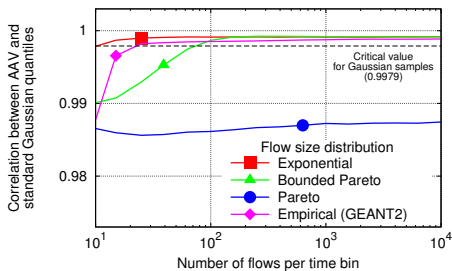

Figure 3: Gaussianity depends on the number of flows and flow size distribution.

4. ALTERNATE ANOMALY DETECTORS

Given the rich literature on anomaly detection methods [2, 12, 13, 26, 29], we need to understand their common features and select a representative set of alternate methods to compare with ASTUTE. The basic approach in previous detection methods is to filter out changes in a traffic time series that are due to normal trends and analyze the remaining residuals. The filtering step can be done using either spatio-temporal information [12, 13, 26, 29] or frequency information [2, 29]. We compare ASTUTE to two methods that use different filtering strategies. We describe these strategies in this section.

We use both of these techniques to find anomalies in five types of metrics computed from traffic traces: (1) *packet counts*, and the entropies of (2) *source IPs*, (3) *destination IPs*, (4) *source ports*, and (5) *destination ports*. While earlier detectors looked for anomalies in time series of packet counts, it was later shown that the entropy time series of IP addresses and ports enables more accurate detections than volume alone [14].

4.1 Kalman: a Spatio-Temporal Detector

The Kalman filter-based method [26] (henceforth, called Kalman) identifies normal traffic changes by learning their correlation structure. Namely, Kalman estimates both spatial (i.e., across different time series) and temporal (i.e., within a single time series) correlations in order to predict the next values of packet counts and entropy. Previous detectors used only purely temporal [12] or purely spatial [13] correlation models. Although there are newer spatio-temporal traffic models [30], we use Kalman in our evaluation because it is simple to parameterize and because, as we discuss below, its threshold parameter has similar semantics to that of ASTUTE, allowing a direct comparison.

Kalman was originally proposed as a network-wide detection technique but, to provide a fair comparison with ASTUTE, we run Kalman using the volume and entropy time series for a single link. We have studied this single-link Kalman by comparing the anomalies it finds with the network-wide version of Kalman [25]. We found that single-link Kalman catches more than 91% of the anomalies found by the network-wide version.

Kalman computes an assessment value (analogous to the AAV from Section 3 but called *innovation*) that follows a Gaussian white noise process if the traffic changes are normally distributed [26]. Thus, given a target false positive rate p, the corresponding Kalman threshold is the percentile $1 - p/2$ of the standard normal distribution $K(p)$, as in ASTUTE (Section 3). This means that a given threshold has the same false positive rates for both ASTUTE and Kalman.

4.2 Wavelet: a Frequency-Based Detector

In the wavelet-based method [2] (henceforth, Wavelet), each time series is decomposed, through wavelet analysis, in low, medium, and high frequency bands. The basic assumption in Wavelet is that, since the low frequency band corresponds to the daily and weekly cycles, these must represent the normal traffic patterns. After the decomposition, the method normalizes both medium and high frequency bands (to have unit standard deviation), and computes a weighted sum of the two signals. Finally, the method computes the variance of the combined signal using a sliding window whose size should match the duration of the anomalies [2]. Wavelet flags an alarm when a set of time bins has variance above a pre-selected threshold T.

In our implementation, we use a Daubechies mother wavelet with four vanishing moments. We combine the medium and high frequency signals with weights of 0.5 each. We use a sliding window of size 30 minutes in order to catch short-lived events such as DoS attacks and port scans.

Unlike with Kalman, there is no well-known relationship between the Wavelet threshold value T and the target false positive rate. This is a major challenge in our evaluation, and we address it with an approach previously used by Zhang et al. [29] to compare methods with different threshold scales. Specifically, for each trace, we pick T so that Wavelet can catch as many anomalies as Kalman does. Since Kalman and Wavelet are both looking for spikes in the same volume and entropy time series, we expect that the top-N Kalman anomalies overlap with the top-N Wavelet anomalies. Note that, choosing T to match the alarms between Wavelet and ASTUTE would not necessarily yield a fair comparison because ASTUTE is *not* looking for volume or entropy spikes, but for violations of its flow independence assumption.

5. EXPERIMENTAL METHODOLOGY

Evaluating anomaly detectors is notoriously difficult. In the absence of ground truth, our community has resorted to two types of approaches. One approach is to inspect each anomaly in order to pinpoint its root cause [2, 14, 20]. However, current root cause analysis practices are largely based on manual analysis, making it error prone and hard to scale for datasets with hundreds of anomalies. Other works have also evaluated their detectors through simulation, either by injecting synthetic anomalies [13, 26, 20] or by replaying real ones that have been manually labeled [14]. The advantage of this approach is that we can vary an anomaly's characteristics, such as its size and duration, and test the detector's sensitivity to these parameters. In addition, simulation is arguably a necessary step in any detector's evaluation [22] since we can replay anomalies in a large enough number of bins to obtain results with high confidence. We employ both manual classification and anomaly injection in order to evaluate ASTUTE as thoroughly as possible.

5.1 Extracting Anomalies from Traffic Data

We analyze traffic traces collected from research and corporate networks. Our traces come from three network links: (1) a link connecting several customers of Internet2 to a backbone router in New York[2]; (2) a peering link between a transit provider and the Frankfurt router at GEANT2[3]; (3) an access link between a corporate site and the rest of a worldwide MPLS enterprise network. In the Internet2 trace, the last 11 bits of all IP addresses are set to zero for host anonymization. Both Internet2 and GEANT2 use Juniper routers, and generate sampled J-Flow statistics at rates of 1/100 and 1/1000, respectively. The corporate network collects non-sampled Cisco NetFlow.

We select these traces to study ASTUTE under a diverse range of potential deployments. The Internet2 trace contains traffic between research institutions that are mostly in the USA. The GEANT2 link comes from a commercial transit provider and thus acts as an entry point from the public Internet to the European NRENs. Both Internet2 and GEANT2 operators care about attacks and outage events that can compromise reachability to their customers. The Corporate trace contains only traffic among hosts inside the enterprise network. All communications with the public Internet are through a proxy located on another site of the network. Although this is a more isolated network environment than those in our other traces, operators still need to manage outages and misconfigured hosts that generate unnecessary traffic in the network.

Table 1 summarizes information about the traces. The table also shows the average utilization of the links. Notice that among our traces, none of the links is running close to 100%. In fact, with a bin width of 5 minutes, no bin exceeded 70% of utilization.

It is worth noting that two of our traces have been collected after random packet sampling. The work of Mai et al. [18] has shown that sampling degrades the performance of different anomaly detectors, leading to both false positives and missed detections. Recall from Section 3.3 that packet sampling does *not* increase the false positive rate in AS-TUTE. Of course, sampling can make low-volume anomalies

Trace	Period	Link utilization
Internet2	Aug 2007	3%
GEANT2	Nov 2007	36%
Corporate	Sep-Dec 2007	0.2%

Table 1: Traffic traces used in our evaluation.

disappear altogether from the trace. This information loss is inherent in sampled data but, as we show in Section 6, our methods are able to uncover a significant number of anomalies in spite of sampling.

When binning the flow records, we assume that packets in a record arrive uniformly spaced in time. This assumption is necessary because flow traces keep timestamps only for the first and last packet in each record, so we cannot know the exact volume of the flow's traffic that should be attributed to each bin. Previous research has shown that this assumption is a reasonable approximation for time bins larger than a few minutes and is particularly more accurate when describing the behavior of long-lived flows [27]. We analyze these traces only at time bins larger than one minute. This assumption only needs to be used for flows that span more than a single bin, and we verified that less than 10% of the flows at a bin width of 1 minute (and less than 1% for bins of 5 minutes) spanned two or more bins. We bin traffic in each trace at 5 minute intervals, since we have observed that non-stationarity violates ASTUTE at timescales longer than that. Previous anomaly detection papers [2, 14, 26] have also used time bins of 5 minutes.

We use a threshold value of 6 for both ASTUTE and Kalman. This threshold value corresponds to a false positive rate of 2×10^{-9} in both methods. We choose this conservative threshold to reduce the number of bins in our manual analysis and to avoid dealing with false positives. For each trace, we set the Wavelet threshold in order to catch as many anomalies as Kalman does, as discussed in Section 4.2. The resulting thresholds in the Internet2, GEANT2, and Corporate traces are respectively 7, 6, and 6.8.

5.2 Manual Classification

We perform root cause analysis for the anomalies found by ASTUTE, Kalman, and Wavelet. However the process of manually classifying hundreds of anomalies is laborious and error-prone. We realized that we could make use of some of the information ASTUTE provides about flows to improve our analysis by making it faster and more reliable. We thus developed a two-step method in which we first use ASTUTE to discover characteristics of the anomalous flows, and then classify the candidate flows by hand in a second step.

5.2.1 Correlated Anomalous Flows

We need to first understand a feature of ASTUTE that makes classification easier; ASTUTE only flags anomalies that contain several flows, and it ignores anomalies that involve only a few flows. Understanding this is important because, if we know that an anomaly exists in a time bin, but ASTUTE is not violated at one of its six flow aggregation levels, we can infer that an anomaly involves only a few flows in that aggregation level.

We simulate anomalies in the GEANT2 trace to measure ASTUTE's sensitivity to the number of anomalous flows and the volume in these flows. To keep it simple, for the moment,

consider that ASTUTE is tracked only for 5-tuple flows, instead of all six flow aggregation levels. First, we identify time bins where there are no ASTUTE anomalies. We do this by computing the AAV for each bin, and keeping the bins with an AAV smaller than 2. For each of these bins, we add different amounts of anomalous traffic. Essentially, we add A anomalous 5-tuples, each with volume change δ_A, for different values of A and δ_A. Then we measure, for each bin, the minimum number of anomalous flows required to trigger an anomaly in ASTUTE. We average this metric across all the bins in the trace.

Figure 4 shows this metric as a function of the volume per anomalous flow, δ_A, and for three different values of the detection threshold. The plot shows that given a threshold value, ASTUTE cannot be violated by fewer than a certain number of 5-tuple flows. Recall from Section 3.1 that when there are few anomalous flows, the average volume change $\hat{\delta}_t$ increases with δ_A at the same rate as the standard deviation $\hat{\sigma}_i$, and the AAV does not exceed $K(p)$.
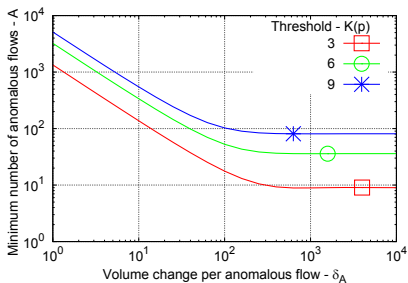
Figure 4: Minimum number of correlated anomalous flows needed to trigger an alarm.

We observe in Figure 4 that, for large values of δ_A, the minimum number of anomalous flows to trigger an alarm converges approximately to the square of the threshold value. We can generalize this lower bound to all threshold values through a simple mathematical model [25]. This relationship between $K(p)$ and the minimum number of anomalous flows is useful for the following reason. If a time bin contains an anomaly that is found by another detector (e.g., Kalman or Wavelet) but not by ASTUTE, then we know that the anomaly involves less than $K(p)^2$ flows. We use this knowledge in our flow identification algorithm in the next subsection.

In the more general scenario, we compute the AAVs for six flow aggregation levels instead of only 5-tuple flows. Many anomalies violate ASTUTE in a subset of the aggregation levels but not in all six of them. The use of multiple aggregation levels is important to identify the flows responsible for an anomaly. For example, consider a source machine scanning several destinations hosts looking for a vulnerable server to attack. The resulting anomaly would generate one 5-tuple flow for each probed target, and thus trigger an AS-TUTE anomaly at the 5-tuple level. However, at the source IP aggregation level, the anomaly is concentrated in a single flow (i.e., the attacker's IP) and, as we have just shown in Figure 4, this does not violate ASTUTE. Once we know that the anomaly lies in a few source IPs it is easier to look for culprits among a few (less than $K(p)^2$) of the largest source IPs, instead of several combinations of small 5-tuples. In

the next section, we formalize this intuition into a method to identify anomalous flows.

Since 5-tuple flows are the finest aggregation level, we should expect that most anomalies that violate ASTUTE at coarser levels (like IPs or ports), should also be found at the 5-tuple level. Indeed, in our traces most of the anomalies are found with 5-tuples only. However, a few anomalies are not found in 5-tuples because they are diluted in a large number of normal flows. At fine flow aggregations like 5-tuples, if the number of correlated flows is too small compared to the number of independent flows, ASTUTE may miss the anomaly. Tracking ASTUTE at six aggregation levels thus also increases the chances of catching an anomaly.

5.2.2 Anomalous Flow Identification

ASTUTE can help with classification by first exposing the set of flows responsible for an anomaly. Once we have identified the responsible flows, the job of manual classification is far simpler since we do not have to search through thousands of possible flows to figure out what has happened. Our flow identification heuristic consists of three steps: (1) we estimate the traffic volume involved in the anomaly; (2) we identify flow aggregation levels where the anomaly can be represented by a few flows; (3) within these aggregation levels, we identify flows whose total volume matches our size estimate. We now detail each of these steps.

Step 1. If an alarm is flagged at any of the six flow aggregation levels where we track ASTUTE, we can estimate the volume of traffic in an anomaly as follows. The confidence interval in Equation (3) is an estimate of the average volume change across all flows in the link. We can multiply the limits of this interval by the number of flows in the link and estimate the total volume of traffic involved in an ASTUTE anomaly. This estimate is given by:

$$I = [\hat{\delta}_i - K(p)\hat{\sigma}_i\sqrt{F},\ \hat{\delta}_i + K(p)\hat{\sigma}_i\sqrt{F}]. \qquad (5)$$

Step 2. If a flow aggregation level L_1 detects an anomaly, but the same time bin does *not* violate ASTUTE at another aggregation level L_2, we can infer that the anomaly must be concentrated in only a few large flows at L_2. Therefore, it is easier to look for candidate anomalous flows in L_2 (where the anomaly is concentrated) than in L_1 (where the anomaly is spread). In the scanning example discussed in the previous subsection, L_1 corresponds to 5-tuple flows and L_2 corresponds to the source IP aggregation level.

Step 3. For the flow aggregations where ASTUTE is *not* violated, we look among the largest flows by size, for a small set of flows whose volume change fits inside the confidence intervals estimated in Step 1. We identify the matching flows as those responsible for the anomaly. If a time bin violates ASTUTE on all six flow aggregation levels, we aggregate flows at the level of network prefixes[4] and check if there is a subnet whose total volume fits one of the estimated confidence intervals. Our rationale behind this is to check for network-level events (e.g., routing changes) that cause particular subnets to appear or disappear at once.

To increase our confidence in the above heuristic, we perform a validation phase. We remove the candidate anoma-

---

[4] Our traces contain prefix information for each flow.

lous flows from the time bin where we found them and check whether ASTUTE is still violated by the remaining traffic.

For anomalies found exclusively by Kalman and Wavelet, we use an alternative heuristic which we discovered to work well in practice. Since these anomalies do not violate AS-TUTE at any of the six flow aggregation levels, the likely explanation is that the anomaly is concentrated on a few large 5-tuples, which we find by inspecting the high-volume 5-tuples in the corresponding time bin.

5.2.3 Anomalous Flow Classification

Based on the flows responsible for a given anomaly, we classify the anomaly by type. To do this, we manually examine the identified flows by looking at features like average packet sizes, application-level protocols, number of sources and destinations, and so on. Table 2 summarizes the criteria we use to label the anomalies found in our three traces. We now discuss each anomaly type in more detail.

Anomalies involving many small packets (e.g., TCP SYNs) usually indicate malicious traffic. We label anomalies as *Denial-of-Service* (DoS) attacks when one or more sources send many small packets to a single destination. When one or more sources send small packets to several destination ports of a single target host, we label it as a *port scan*.

Other anomalies are caused by unusual application behavior. In all traces, we found anomalies due to *large file transfers*, i.e., hosts that download files from one or more sources at the same time. We differentiate those transfers from DoS attacks by two features: (1) bulk data transfers typically use large TCP packets (above 1024 bytes); and (2) the sources often use well-known application protocols such as HTTP or FTP.

In the corporate trace, we found anomalies related to a series of *misbehaving applications*. For instance, we observed that once a day at 2:00 AM (local time), hundreds of hosts simultaneously broadcast name service requests. This anomalous behavior is caused by a default setting in the name server used within the enterprise. We have notified the IT operators of the corporate network (who were unaware of the anomaly). A similar example is an instant messaging application used between employees. In this anomaly, we found a number of clients continuously trying to establish TCP connections to a non-responsive (and probably unreachable) server. Our anomaly detector could have been used to warn the operators of this problem.

We found anomalies involving flows that share nothing in common except that all source (or destination) IPs are contained in one or a few subnets. After inspecting those anomalies, we noticed gaps in the traffic associated with those subnets' prefixes and labeled them as *prefix outages*. These anomalies may be associated with routing changes or also upstream/downstream link outages that cut the reachability to the corresponding networks. Many of these prefix outages were in GEANT2, and most were associated with a research institution in Greece whose prefix route kept flapping several times in one day.

We also found that many anomalies were caused by trace gaps of at least 10 seconds. Such gaps impact *all* of the link's traffic, and because of that, they violate ASTUTE in all six flow aggregation levels. If a gap lasts more than a full time bin (i.e., 5 minutes), we assume that it is due a persistent *link outage*. Otherwise, we assume that it is simply a *measurement gap*. Measurement gaps are caused by a bug in one of the implementations of J-Flow (Juniper's equivalent of NetFlow) that has been recently documented in the literature [6]. Since J-Flow is used in both the Internet2 and GEANT2 networks, we see such anomalies from those networks in our data. The corporate trace is collected with Cisco NetFlow and does not exhibit such gaps.

Using these criteria, we have classified a variety of events, ranging from abuse traffic (e.g., DoS attacks and scans) to operational problems (e.g., outages and misconfigured enterprise applications). Even though these are all statistical anomalies, not all operators care about their root cause events. For example, a transit ISP may care about outages and routing disruptions but not about DoS attacks (if its links are over-provisioned to avoid congestion). When evaluating an anomaly detector, it is important to differentiate the definitions of (a) statistical anomalies and (b) events that an operator cares about. We focus on the definition of statistical anomalies to evaluate our detectors. Accordingly, we define false positives in the statistical sense, i.e., time bins that follow the normal traffic assumptions (e.g., weakly correlated flows for ASTUTE) but are flagged as anomalous.

5.3 Simulation through Anomaly Injection

Simulation helps us understand how methods trade-off detection rates for false positives. We use this particular type of evaluation to compare Kalman and ASTUTE only. Recall from the discussion in Section 4.2 that the relation between the Wavelet detection threshold and its false positive rate does not follow the same theoretical argument shared by Kalman and ASTUTE. In fact, we are not aware of a theoretical relationship between the Wavelet threshold and its expected false alarm rate [2].

Our method for anomaly injection consists of three steps. First, we build a set of benchmark anomalies for injection. Then, we inject these anomalies, one at a time, in each non-anomalous bin and measure how much they impact both the AAV and the Kalman innovation. Finally, for a given method and a threshold value, we measure the method's *detection rate* for each type of anomaly.

Our benchmark set contains the same types of anomalies as those shown in Table 2. Since we have manually identified the flows responsible for several ASTUTE and Kalman anomalies (Section 5.2.3), we can recreate those anomalies by adding their flows to other time bins. This makes our experiment realistic since (1) our benchmark contains real anomalies, and (2) we are sure that the IPs involved in these anomalies (i.e., attackers, victims, and misbehaving hosts) actually have their traffic routed through the link where we want to inject them.

However, we can only inject the manually identified anomalies that are associated with end-host activity, i.e., DoS attacks, port scans, large file transfers, and misbehaving enterprise applications. Anomalies caused by prefix or link outages (or other network problems), impact a different set of flows depending on the bin where they happen. We simulate prefix and link outages in a given time bin by removing the subset of the bin's traffic that is associated with a known anomalous prefix in the former case, or all the prefixes in the latter case. In addition, we can re-create these outages at different durations to learn how large an outage needs to be to be accurately detected.

We have also considered simulating a type of anomaly which we did not observe in our traces: namely, congestion

Anomaly class	Description
DoS attack	Many small packets from one or more source IPs to a single destination IP.
Port scan	Small packets from one source IP to several ports in one destination IP.
Large file transfer	One or more TCP bulk flows (i.e., large packets) to a single destination IP.
Misbehaving apps.	Multiple causes; see text in Section 5.2.3.
Prefix outages	Intervals where the traffic of one or more prefixes (but not all of them) disappears.
Link outages	Intervals larger than five minutes with no flow arrivals.
Measurement gaps	Intervals smaller than five minutes with no flow arrivals.
Unknown	Anomalies for which we could not identify their root causes.

Table 2: Criteria used to classify events based on characteristics of the anomalous flows.

in upstream links. However, for the purposes of our simulation, the impact of upstream congestion should be similar to that of the prefix outages described above, i.e., we should observe a correlated decrease in the throughput from flows that cross the congested link. Since ASTUTE detects prefix outage anomalies, we conjecture that it should also flag alarms due to upstream congestion if a significant number of the flows experiencing congestion cross a link where ASTUTE is deployed.

Using this injection methodology, we are able to plot ROC curves, which show detection rates as a function of false positive rates. Given a trace and an anomaly type, we vary the detection threshold for ASTUTE and Kalman (top axis) and compute both the false positive rate (bottom axis) and detection rates (left axis). The false positive rate comes from the theoretical model that both the AAV and the Kalman innovation follow a standard normal distribution in the absence of anomalies (Sections 3 and 4.1). For a fixed threshold value, the detection rate for either method is the fraction of injections that trigger a detection in the method.

6. PERFORMANCE EVALUATION

Our results essentially show that ASTUTE finds a new family of anomalies. While Kalman and Wavelets tend to find anomalies involving few large flows (e.g., DoS attacks), ASTUTE is much more accurate at finding anomalies involving several small flows (e.g., scans and outages). In addition, we show that this result is independent of threshold values.

6.1 Anomaly Characteristics

Table 3 shows the number of anomalies found by each method in each trace. We measure the intersection between anomalies found by ASTUTE and anomalies found by the other two methods. We also show the overlap between Kalman and Wavelet, and the total number of anomalies found by all methods together. Our main observation is that the overlap between ASTUTE and the other methods is small: only 6% of all the anomalies, and less than 30% of the combined detections of Kalman and Wavelet. This suggests that ASTUTE can complement the detections from the other two methods, which are based on volume and entropy. In addition, Kalman and Wavelet have more overlap among each other than with ASTUTE. Note that this is not simply a matter of our choice of thresholds; if we increase the ASTUTE threshold so that it detects the same number of anomalies as Kalman and Wavelet the new overlap could only be smaller since ASTUTE would flag fewer alarms.

To understand the differences between anomalies found by each method, we inspected and classified the cause of

Anomaly Set	Internet2	GEANT2	Corporate
ASTUTE ($\mathcal{A}$)	351	99	61
Kalman ($\mathcal{K}$)	56	24	16
Wavelet ($\mathcal{W}$)	56	24	16
$\mathcal{A} \bigcap (\mathcal{K} \bigcup \mathcal{W})$	19	12	3
$\mathcal{K} \bigcap \mathcal{W}$	49	19	7
$\mathcal{A} \bigcup \mathcal{K} \bigcup \mathcal{W}$	395	116	83

Table 3: Anomalies found by each method.

each alarm. Using the heuristics described in Section 5.2, we are able to identify the anomalous flows for 59% of all the ASTUTE anomalies. We found that the remaining 41% of ASTUTE anomalies are caused by measurement gaps or by link outages which essentially impact *all* flows in the link. We verified this by measuring the flow inter-arrival times in the corresponding anomalous bins, and noting that each of these bins contain gaps of at least 10 seconds without any flows. We found that gaps this long are unusual, and do not occur in other anomalous bins, nor in the normal bins. For these events, we define the anomalous flows as those that are impacted by the outage, i.e., flows that exist before and after, but not during the gap. Using the procedures described in Section 5.2, we are able to identify the flows causing all anomalies in our traces.

Table 4 summarizes the number of anomalies per type found in each trace and by each method. Note that ASTUTE is more effective at detecting anomalies such as port scans, prefix outages, misbehaving enterprise applications, and even sometimes link outages (in the case of the corporate network) than the Kalman and Wavelet methods. However, Kalman and Wavelet are better at detecting DoS attacks in both Internet2 and GEANT2.

We also observe that Kalman and Wavelet find more large file transfers than ASTUTE in the corporate network, but the situation is reversed in Internet2 and GEANT2. All of the file transfers found in the corporate trace involve few local hosts communicating with the company's gateway to the public Internet (which is located on another corporate site and thus the transfers are seen in the MPLS link). Conversely, the file transfers found in Internet2 and GEANT2 usually involve a single destination host, receiving traffic from several (up to hundreds) of sources simultaneously (due to peer-to-peer transfers).

Finally, we could not discover the root causes of five anomalies (out of 600); we labeled them as *unknown* anomalies. These anomalies may be either due to false positives (which

Internet2				
Anomaly type	Total	$\mathcal{A}$	$\mathcal{K}$	$\mathcal{W}$
DoS attack	38	1	37	30
Port scan	198	198	0	2
Large file transfer	2	2	0	0
Prefix outage	1	1	0	0
Link outage	15	12	12	15
Measurement gap	136	135	4	6
Unknown	5	2	3	3
Total	395	351	56	56

GEANT2				
Anomaly type	Total	$\mathcal{A}$	$\mathcal{K}$	$\mathcal{W}$
DoS attack	16	0	16	12
Port scan	8	8	4	5
Large file transfer	4	4	0	0
Prefix outage	37	36	3	7
Measurement gap	51	51	1	0
Total	116	99	24	24

Corporate				
Anomaly type	Total	$\mathcal{A}$	$\mathcal{K}$	$\mathcal{W}$
Large file transfer	20	0	13	11
Misbehaving apps.	38	37	1	1
Link outage	12	11	2	4
Prefix outage	13	13	0	0
Total	83	61	16	16

Table 4: Anomaly types found on each trace by AS-TUTE ($\mathcal{A}$), Kalman ($\mathcal{K}$), and Wavelet ($\mathcal{W}$).

Figure 5: Map of the traffic anomaly spectrum.

is unlikely, given the high threshold we use) or simply real anomalies that we cannot explain through visual inspection.

Figure 5 maps the qualitative properties of the anomalies (i.e., their labels) to their quantitative properties (i.e., their number of flows and packets). Each point corresponds to all the anomalies of a given type (represented by the point shape) that are detected in a given network (represented by the point fill pattern) by any of the three methods. Each point's coordinates correspond to the number of anomalous 5-tuple flows (x axis) and the average anomalous flow size in packets (y axis). The coordinate values are averaged across all the anomalies represented by the corresponding point.

Figure 5 characterizes the different types of anomalies detected by ASTUTE, Kalman, and Wavelet. We draw vertical and horizontal lines to divide the plot in quadrants that isolate the types of anomalies in which ASTUTE is better than Kalman and Wavelet, and vice-versa. The top-left and bottom-right quadrants respectively correspond to anomalies with a few large flows and to anomalies with many small (or medium-sized) flows. The bottom-left quadrant contains no anomalies since it represents "well-behaved" traffic applications, i.e., few flows of moderate size. The top-right quadrant is also empty since every link has a finite capacity and it is very unlikely to find many large flows at once in a link. All the anomalies for which Kalman and Wavelet perform better than ASTUTE in Table 4 lie in the top-left quadrant of the plot. Conversely, ASTUTE detects more anomalies

than Kalman and Wavelet in all points represented in the bottom-right quadrant of the plot.

In addition to the difference in anomaly types, Figure 5 shows that ASTUTE finds anomalies that often involve less total volume than anomalies found by Kalman and Wavelet. Note that if we multiply the x and y coordinates of points in Figure 5, we obtain the average total volume in an anomaly type. We add diagonal lines in the plot to show different orders of magnitude of aggregate volume in packets. For each network, note that the anomaly types for which ASTUTE has more detections (bottom-right) usually involve one or two orders of magnitude less volume than those types where Kalman and Wavelet are better. For instance, while the DoS attacks in GEANT2, found by Kalman and Wavelet, typically involve hundreds of thousands of packets (per 5-minute bins at 1/1000 sampling), the port scans found by ASTUTE in the same network involve only thousands of packets.

This pattern can be observed across the three traces and it confirms that ASTUTE cannot detect anomalies caused by a few large flows while it is very sensitive to anomalies involving many small flows. Since both types of anomalies are important, we do not advocate that operators should use ASTUTE as a replacement for detectors based on volume and entropy. However, our findings suggests that either Kalman or Wavelet, combined with ASTUTE, cover more exhaustively the space of traffic anomalies.

6.2 Detection Performance

In this section, we use simulation (described in 5.3) in order to understand how ASTUTE and Kalman trade-off detection rates for false positives. Our analysis shows that the main result from the previous section, namely that ASTUTE and Kalman are good at different types of anomalies, also holds for different threshold values. Figure 6 shows ROC curves for four anomaly types in GEANT2. The plots for other anomaly types and other traces contain similar curves; we omit them for conciseness.

The ROC plots in Figure 6 display three different behaviors in the relative performance of ASTUTE and Kalman. First, plot 6(a) shows that even though Kalman can detect most port scans, ASTUTE is much more sensitive to these anomalies, particularly for threshold values above 3. Second, plots 6(b) and 6(c) show that ASTUTE exposes short outages (30-second link outages and 2-minute prefix outages in 5-minute bins) that Kalman can barely detect at all. Al-
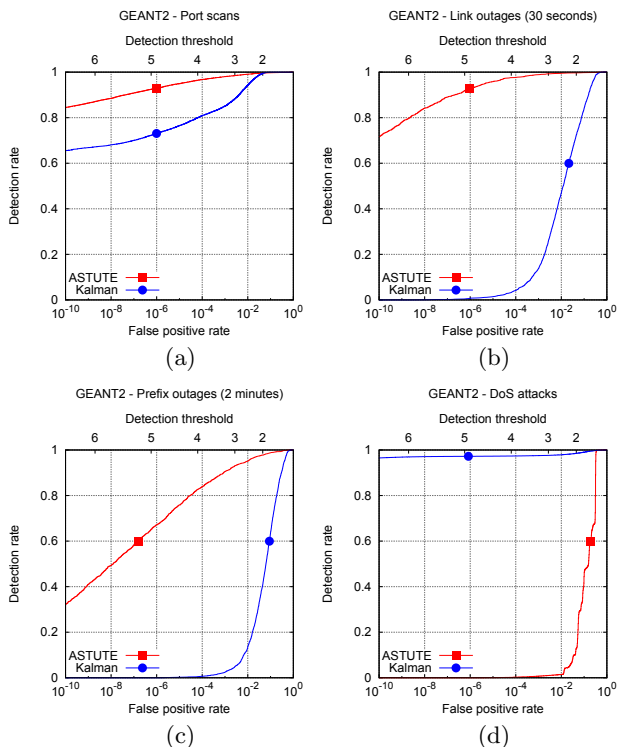
Figure 6: ROC curves comparing ASTUTE and
Kalman for different anomaly types in GEANT2.
False positive rates are based on statistical assump-
tions (described in Sections 3.3 and 4.1).

though a single short outage has limited impact on end-user
performance, a large number of alarms due to outages can
be an indication of faulty links [19] or routing misconfigu-
rations. Third, plot 6(d) shows that ASTUTE misses DoS
attacks which Kalman can easily spot. Recall from Figure 5
that the DoS attacks in the GEANT2 trace tend to involve
only a few large flows, which do not violate ASTUTE (as
shown in Section 5.2.1).

6.3 Complementarity of ASTUTE and Kalman

Fixing the detection thresholds for ASTUTE and Kalman,
we compute the detection rate of a detector that combines
alarms from both methods. To determine the combined false
positive rate, we consider lower and upper bounds as follows.
Let the false positive rates from ASTUTE and Kalman be
f_A and f_K, respectively. In the best case, bins that trigger
false alarms in one method also do so in the other method,
and thus the combined false positive rate is the maximum
between f_A and f_K. Conversely, if ASTUTE and Kalman
trigger false alarms in different bins, then the combined false
positive rate is either $f_A + f_K$ or 1, whichever is smaller.

In order to compute a single ROC curve for different
classes of anomaly, we need to decide the relative frequency
of events in each class, which we call the *anomaly mix*. Since
we cannot know a priori the anomaly mix in a trace, we com-
pute our ROC curves under two scenarios. First, we consider
that the frequency of each anomaly type is proportional to
the number of such anomalies *observed* in a trace, accord-
ing to Table 4. In the second scenario, we consider that the
frequency of anomalies is *uniform* across all classes.

Figure 7 shows ROC curves for ASTUTE, Kalman, and
the combined detections in GEANT2, assuming equal thresh-
old values for ASTUTE and Kalman. We plot ROC curves
of the combined detections using lower and upper bounds for
the false positive rate as described above. The plots show
that the two methods combined perform better than each
by itself, even in the worst case, when the false alarms from
ASTUTE and Kalman add up. For example, in the observed
anomaly mix (left plot in Figure 7) for a 1% false positive
rate, ASTUTE alone misses 15% of the anomalies, Kalman
misses 35%, but the combined methods only miss 5%. The
corresponding ROC curves for the other two traces show the
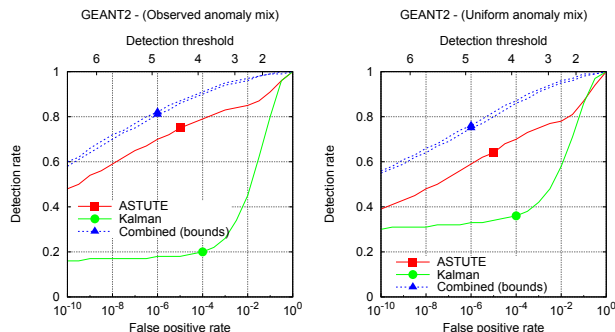same qualitative result and we omit them here.

Figure 7: ROC curves for ASTUTE, Kalman, and
their combined detections in GEANT2.

7. RELATED WORK

Significant attention has been devoted to anomaly detec-
tion in recent years. Early anomaly detection techniques re-
lied only on volume metrics such as packet and byte counts [2,
13, 26]. Lakhina et al. [14] showed that the entropy of fea-
ture distributions (e.g., IP addresses and ports) extends the
set of detections by volume metrics. More recently, Ny-
chis et al. [20] have shown that the entropies of flow size
and degree distributions can flag low-volume anomalies in
their dataset that go unnoticed in the entropies of addresses
and ports. Note, however, that their work is concerned with
new entropy-based *metrics* for anomaly detection. Such new
metrics can, in principle, be used with any detector that
learns their normal trend from time series data and flags
bins which deviate from that trend. On the other hand, we
develop a new *method* which is fundamentally different from
time series approaches used by Kalman [26], Wavelet [2], or
PCA [13], because it does not require training from normal
traffic data, and thus is immune to data-poisoning.

There are specific anomaly detection methods for finding
network disruptions such as link or prefix outages. For ex-
ample, some methods can detect routing-induced anomalies
by analyzing BGP feeds [11] or by combining those feeds
with traffic data [23]. Our method is capable of finding this
type of anomalies looking only at traffic data by detecting
the correlated increase or decrease in the throughput of the
impacted flows.

Tangentially related to our work is recent research that
has addressed the complementary problem of automated
root cause analysis of traffic anomalies [17, 3, 24] which typ-
ically involves identifying the flows involved in the anomaly
and classifying their behavior.

8. CONCLUSIONS

We presented ASTUTE, a new method for traffic anomaly detection in network links. The main novelty in ASTUTE's design is that, unlike traditional anomaly detectors, it does not need to learn normal traffic behavior from traffic traces. Instead, ASTUTE relies only on empirical traffic properties which hold for highly aggregated network links.

We showed that *ASTUTE uncovers anomalies that are qualitatively and quantitatively different than those found by two other methods* from the literature: Kalman [26] and Wavelet [2]. The anomalies found by ASTUTE are caused by groups of flows that simultaneously increase or decrease their traffic, even if the traffic in these flows is small compared to the total link traffic. This result is supported by experiments with real traffic traces, including manual classification and simulation through anomaly injection.

We showed that ASTUTE cannot find large volume anomalies caused by one or a few flows. These types of anomalies are easily found by previous techniques such as Kalman and Wavelet. Our simulations showed that *ASTUTE and Kalman complement each other nicely*. This suggests that a hybrid detector based on ASTUTE and one of the previous techniques is likely to cover a broad spectrum of anomalous traffic. We will explore this idea in our future work.

Besides detection, *ASTUTE also provides information that is useful to perform root cause analysis*. After having manually identified hundreds of anomalies in our three traces, ASTUTE made it easier to determine the type of each anomaly and to identify the responsible set of flows. We plan to explore how to automate the heuristics based on ASTUTE as a method for classifying anomalies.

Finally, *ASTUTE's computational cost is fairly small*, and smaller than detectors that require training phases [25]. Besides the process of binning flows (needed for all detectors), it takes 21 seconds to calibrate Kalman on the GEANT2 trace, while ASTUTE is computed in less than a second.

Acknowledgements

9. REFERENCES

[1] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. A Flow-Based Model for Internet Backbone Traffic. In *Proceedings of IMW*, 2002.

[2] P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *Proceedings of IMW*, 2002.

[3] D. Brauckhoff, A. Wagner, X. Dimitropoulos, and K. Salamatian. Anomaly Extraction in Backbone Networks using Association Rules. In *Proceedings of IMC*, 2009.

[4] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. On the Nonstationarity of Internet Traffic. In *Proceedings of SIGMETRICS*, 2001.

[5] M. E. Crovella, M. S. Taqqu, and A. Bestavros. Heavy-Tailed Probability Distributions in the World Wide Web. In R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors, *A Practical Guide To Heavy Tails*, chapter 1, pages 3–26. Chapman and Hall, 1998.

[6] I. Cunha, F. Silveira, R. Oliveira, R. Teixeira, and C. Diot. Uncovering Artifacts of Flow Measurement Tools. In *Proceedings of PAM*, 2009.

[7] D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer-Verlag, 1988.

[8] W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, 3rd edition, 1968.

[9] J. J. Filliben. The Probability Plot Correlation Coefficient Test for Normality. *Technometrics*, 17(1):111–117, 1975.

[10] N. Hohn, D. Veitch, and P. Abry. Cluster Processes, a Natural Language for Network Traffic. *IEEE Transactions on Networking*, 51(8):2229–2244, 2003.

[11] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu. Diagnosing Network Disruptions with Network-Wide Analysis. In *Proceedings of SIGMETRICS*, 2007.

[12] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-Based Change Detection: Methods, Evaluation, and Applications. In *Proceedings of IMC*, 2003.

[13] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *Proceedings of SIGCOMM*, 2004.

[14] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies Using Traffic Feature Distributions. In *Proceedings of SIGCOMM*, 2005.

[15] E. Lehmann and J. Romano. *Testing Statistical Hypotheses*. Springer, New York, NY, USA, 2006.

[16] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic. *Transactions on Networking*, 2(1):1–15, 1994.

[17] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and Identification of Network Anomalies using Sketch Subspaces. In *Proceedings of IMC*, October 2006.

[18] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is Sampled Data Sufficient for Anomaly Detection? In *Proceedings of IMC*, 2006.

[19] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of Failures in an IP Backbone. In *Proceedings of INFOCOM*, 2004.

[20] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An Empirical Evaluation of Entropy-based Anomaly Detection. In *Proceedings of IMC*, 2008.

[21] V. Paxson and S. Floyd. Wide Area Traffic: the Failure of Poisson Modeling. *Transactions on Networking*, 3(3):226–244, 1995.

[22] H. Ringberg, M. Roughan, and J. Rexford. The Need for Simulation in Evaluating Anomaly Detectors. *SIGCOMM CCR*, 38(1):55–59, 2008.

[23] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman. Combining Routing and Traffic Data for Detection of IP forwarding anomalies. In *Proceedings of SIGMETRICS*, 2004.

[24] F. Silveira and C. Diot. URCA: Pulling out Anomalies by their Root Causes. In *Proceedings of INFOCOM*, 2010.

[25] F. Silveira, C. Diot, N. Taft, and R. Govindan. ASTUTE: Detecting a Different Class of Traffic Anomalies (Extended Version). Technical report, Technicolor, 2010.

[26] A. Soule, K. Salamatian, and N. Taft. Combining Filtering and Statistical Methods for Anomaly Detection. In *Proceedings of IMC*, 2005.

[27] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger. A Methodology for Studying Persistency Aspects of Internet Flows. *SIGCOMM CCR*, 35(2), 2005.

[28] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. *Transactions on Networking*, 5(1):71–86, 1997.

[29] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network Anomography. In *Proceedings of IMC*, 2005.

[30] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-Temporal Compressive Sensing and Internet Traffic Matrices. In *Proceedings of SIGCOMM*, 2009.