Influence of Data-Reduction Techniques on Traffic Anomaly Detection

Paper #232, 14 pages

ABSTRACT

Statistical techniques for detecting anomalous traffic can be an invaluable tool for the operators of large IP networks. However, the effectiveness of anomaly-detection schemes is extremely sensitive to the data-reduction methods used to manage the large volume of data and identify the statistical outliers. In this paper, we analyze the impact of sampling, temporal aggregation, and IP address anonymization on anomaly detection, focusing on one week of data for the Abilene and Geant backbones. In contrast to previous work, our evaluation methodology allows us to study two important metrics—the false-positive rate and the anomaly type—that are crucial for a meaningful evaluation. We find that, although Abilene and Geant differ substantially in the number and type of anomalies, they show similar trends for the effects of data-reduction techniques. All of the data-reduction methods reduce the number and diversity of anomalies the statistical techniques can detect. In addition, sampling introduces extra false positives by making the data more "spiky," temporal aggregation can sometimes merge multiple anomalies into a single time bin, and IP address anonymization can sometime help in detecting IP scans. Our results are an important step toward helping network operators make informed trade-offs between the anomalies they wish to detect and the system overheads they must endure.

1. INTRODUCTION

Traffic anomalies, such as flash crowds, denial-of-service attacks, port scans, and the spreading of worms, can have detrimental effects on Internet services. Detecting and diagnosing these anomalies is critical to network operators, who must take corrective actions to alleviate congestion, block attacks, and warn affected users. Anomaly detection requires digging into massive amounts of measurement data, which is a task best left to automated analysis. A common approach to this problem has been to discretize network traffic into timeseries, which are analyzed by statistical-analysis techniques, and to equate detected outliers with traffic anomalies. Before such analysis can be performed, however, the operators apply *data-reduction* techniques—such as sampling packets or aggregating in time or space—because collecting, storing, and analyzing packet-level traces at line rate from many vantage points in the network is impractical. This paper investigates the impact that these different data-reduction techniques have on traffic anomaly detection.

The anomaly-detection pipeline depends on a number of tunable parameters, such as the time scale at which IP flows are binned, the packet sampling rate, the number of bits anonymized of IP addresses, the network-wide representation (e.g., traffic matrix), the statistical-analysis technique used to find outliers, and how this technique is tuned (e.g., detection threshold). In this paper, we investigate the influence of temporal aggregation, packet sampling, and IP address anonymization on the effectiveness of Kalman filter-based traffic anomaly-detection for a range of detection thresholds. Our study yielded over 100 thousand combinations of parameter settings, 200 million timeseries data points, and 15 million statistical anomalies.

In order to gain a deeper understanding of the influence of the tunable parameters, we need to analyze more meaningful metrics than the number of statistical outliers. Ultimately we wish to know how each parameter impacts our ability to detect various kinds of underlying network events, e.g. flash crowds, traffic shifts, and DoS attacks. This is the distinction between a *statistical anomaly* and a *traffic anomaly*, and it is the latter that gives us access to important metrics such as the *false-positive rate* and the *anomaly type*—i.e., does a given statistical anomaly correspond to an anomalous underlying network event, and what type of event is it.

Unfortunately, the root-cause analysis necessary to identify the specific traffic anomaly associated with a given statistical anomaly is simultaneously too time-consuming to be done manually and presently beyond the capability of automated techniques. We therefore need to reach a compromise between statistical and traffic anomalies that gives us access to meaningful notions of both ground truth and anomaly type, but can be determined efficiently. Our approach therefore leverages humans' ability to readily identify the start and end of a sequence of detected statistical anomalies, which falls into an *anomalous region* that is distinct from mere noise. In addition to providing a false-positive rate, the anomalous region can be characterized in terms of its statistical properties—e.g., whether it corresponds to an increase or decrease in packet counts. We have found that this procedure provides rich insights into the influence of the parameters while also being efficient.

The software that implements our methodology consists of two main components. The first component automatically parses hundreds of gigabytes of flow traces, sweeps our parameter space, constructs entropy timeseries, applies a Kalman-filter based anomaly detector, and outputs databases with records for the entropy timeseries and statistical anomalies. Since efficiency is clearly imperative, this component was implemented as a parallelized program and run on a computer cluster. The second component—WebClass—is a web-based tool that parses these databases and allows human operators to interactively label a sequence of statistical anomalies as belonging to an anomalous region or being a false positive. After a user has identified an anomalous region, WebClass automatically determines its type, before everything is stored back in the database along with the id of the operator who classified the region. WebClass is designed for concurrent usage, which has allowed us to classify nearly 400 thousand statistical anomalies. We intend to release our entire system and resulting traces to the research community.

Applying our software to one week of network-wide IP flow traces from both the Abilene and Geant backbone networks, we show that the tunable parameters have a significant impact on the effectiveness of traffic anomaly detection. We find that Abilene has a much lower false-positive rate and a large fraction of anomalous regions correspond to very few anomaly types, which we argue is consistent with their respective traffic mixes. Moreover, we find that temporal aggregation, sampling, and IP address anonymization have similar impacts on both networks. In particular, temporal aggregation and sampling not only reduce the total number of statistical anomalies, but also the variety in anomalies that can be detected because the most prevalent anomalies are often the least affected by aggregation. Aggressive sampling also significantly increases the false-positive rate.

The remainder of this paper is organized as follows: section 2 discusses related work; section 3 describes our measurement data and our software for automatically creating a traffic anomaly database; section 4 describes how our Web-based tool is used to identify false-positives and the anomaly type; section 5 describes and compares the Geant and Abilene networks using these metrics while sweeping the Kalman detection threshold; sections 6, 7, and 8 analyze the impact of varying the temporal aggregation, IP anonymization, and sampling, respectively; finally, section 9 contains our conclusions.

2. RELATED WORK

Traffic anomaly detection has received a great deal of attention in the research literature. While there has been some work that leverages clever data structures to find heavy-hitters [1, 2], most papers have utilized statistical-analysis techniques to detect outliers in traffic timeseries. Numerous techniques have been evaluated, including wavelets [3], moving average variants, Fourier transforms [4, 5], Kalman filters [6], and PCA [7]. Early work in this area often analyzed data from a single link [3], whereas more recent papers have shown very promising results by analyzing network-wide measurements [8].

With such a large body of work, it becomes increasingly important to be able to compare and contrast presented approaches. While there have been a few papers that compared a subset of the statistical-analysis techniques [4, 5], researchers have only very recently begun investigating how data-reduction techniques impact the ability to detect traffic anomalies [9]. Much in the same way that early papers on traffic anomaly detectors had a limited scope, this new line of work has analyzed the impact of only one form of data-reduction [10], on only one type of traffic anomaly [11], or analyzed data from a small number of links [12]. This paper investigates the impact of varying three important data-reduction techniques and the detection threshold on the ability of a network-wide traffic anomaly detector to effectively detect a large variety of anomalies.

3. CREATING AN ANOMALY DATABASE

This section describes our methodology and software for building a traffic anomaly database. We detail the two backbone networks whose IP flow traces were thinned according to our parameter space (3.1), aggregated into traffic matrices (3.2), transformed into entropy timeseries (3.3), and analyzed by a Kalman filter in order to identify statistical anomalies (3.4). We will end with a description of our implementation of this methodology (3.5).

3.1 Traffic Measurement Data

We use a full week of IP flow traces collected between November 21st and 27th, 2005, for both the Abilene [13] and Geant [14] backbone networks. Both networks collect their flow statistics using Juniper's J-Flow tool [15]. Abilene is an 11-node research backbone that connects Internet2 universities and research labs across the continental United States. Abilene does not provide transit services to the Internet at large. Instead, its customers must maintain separate connections to the commodity Internet [16]. Geant is a 23-node network that connects national research and education networks representing 30 European countries. Unlike Abilene, Geant does provide Internet connectivity to its customers.

Network	Nodes	Sampling	Time Agg	Anon
Abilene	11	1%	5 min	11 bits
Geant	23	0.1%	15 min	0 bits

Table 1: Networks studied

As in many backbone networks, the operators of Abilene and Geant apply data-reduction techniques in collecting and storing their measurement data. IP flow traces are discretized into timeseries of measurement values (e.g. packet counts), which allow for efficient processing by statistical-analysis techniques. Abilene has chosen to aggregate its IP flows into 5-minute time bins, compared to 15-minute windows for Geant. Packet sampling is another very common data-reduction technique. Abilene randomly samples 1 out of every 100 packets for inclusion in the flow statistics whereas Geant samples packets at 1 out of 1000. Finally, Abilene anonymizes the eleven least-significant bits of IP addresses in flow records in order to protect user privacy, which also reduces the volume of measurement data. Geant performs no such anonymization of IP header fields. These default data-reduction levels are summarized in table 1.

We evaluate the impact of these data-reduction techniques by taking them further. Further temporal aggregation occurs by combining adjacent time bins and further anonymization is equally straightforward. Subsampling is only slightly more complicated in that, for every IP flow record that was said to have contained n packets, we simulate sampling each of the n packets at the given sampling rate. If a flow has 0 packets after subsampling then it is removed from the flow trace. The exact parameter values that we chose to explore will be detailed in the result sections, but the end result of these steps is a further thinned IP flow trace. We clearly cannot analyze finer aggregation levels or lower sampling rates than the operational networks originally used in collecting the data.

3.2 Computing the Traffic Matrix

To effectively detect traffic anomalies, the network operators must aggregate IP flows into a representation that is computationally manageable and reveals underlying spatial and temporal trends. We represent the data as a traffic matrix by aggregating the measurement data based on where the traffic entered and left the network (aka OD-flows). Identifying the ingress router i for a given IP flow is straightforward because both Abilene and Geant have separate traffic logs for each ingress router in their respective networks. Identifying the egress router e can be more cumbersome, however; performing this identification requires routing information from the relevant network, the ingress router, the destination address, and the destination IP address mask. The latter two pieces of information are embedded in the IP flow records and we've already identified the ingress router, which leaves only parsing of the routing data.

The purpose of parsing the routing data is to acquire the mapping between every $\langle i, p \rangle$ combination (where p is the masked destination IP address) and the associated egress point e. For Abilene it is sufficient to parse only the BGP records exported by its Zebra BGP monitors in order to acquire this mapping. Geant, on the other hand, has one Zebra BGP monitor embedded in an iBGP mesh that logs BGP records that identify a set of possible egress points E for every prefix. One must therefore also parse Geant's IS-IS logs in order to find the router $e \in E$ that has the minimum-cost path from i, which will give us the mapping, and traffic matrix we seek. Therefore, the contents of each $\langle i, e \rangle$ cell in the traffic matrix is, at this stage, all the IP flow records for the traffic that entered the network at router i and exited at router e.

3.3 Computing Entropy Time Series

Previous work [7] has demonstrated that traffic anomaly detectors that analyze entropy timeseries of the four main IP header features (source IP address, destination IP address, source port, and destination port) can be quite effective. Entropy is used because it provides a computationally efficient way to measure the dispersion or concentration in a distribution, and a wide variety of anomalies will manifest themselves as a shift in the distribution of one or more of these IP features. That is, for every $\langle i, e \rangle$ cell in the traffic matrix, we compute the entropy values of the four IP header features for all the traffic that passed between ingress router i and egress router e. The entropy of a random variable X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} Pr(X = x_i) \log_2 Pr(X = x_i) \qquad (1)$$

where $Pr(X = x_i)$ is the probability of event $x_i \in X$ occurring. In our context, the events are observations of a given IP feature. For example, the probability of seeing port 80 is defined to be the number of sampled packets using port 80 divided by the total number of packets in the given time interval. A sudden flash crowd to a Web-server will therefore cause a specific destination IP address (the Web-server) and destination port (port 80) to become much more prevalent than in previous time steps, which will cause a decrease in the destination IP address and destination port entropy timeseries, respectively. A more complete explanation of the benefits of using entropy for traffic anomaly detection can be found in [7].

3.4 Applying Kalman Anomaly Detector

After having constructed timeseries of entropy values, one must use a statistical-analysis technique in order to detect outliers—statistical anomalies. We have chosen to use a traffic anomaly detector based on Kalman filters [6] because its performance has compared favorably to other prominent statistical-analysis techniques [8]. The relevant Matlab code was written by Soule *et al.* [6]. At the high level, the algorithm takes a traffic matrix F (corresponding to entropy values for only one of the IP header fields) and detection threshold T_h as input, and returns a boolean for every point in the traffic matrix, which indicates whether the given IP header feature of the given OD flow at the given point in time was classified as a statistical outlier.

The anomaly detection method based on a Kalman filter has three primary steps. The first stage is a calibration step in which an Expectation-Maximization algorithm is used to find the best linear model to fit the data (e.g., a traffic matrix) as described in [17]. Second, this model is utilized by the Kalman filter as detailed below. The final step is to detect statistical outliers by comparing the difference between the filtered and the observed values, which are a part of the Kalman filter procedure.

The Kalman filter is itself composed of two steps applied sequentially as soon a new measurement is available for analysis. The first step predicts the entropy value at time $t + 1$ based on previously observed values up to time t. The second step estimates the value at time $t + 1$ based on the measured value at time $t + 1$ and the predicted value for time $t + 1$.

Prediction Step The prediction of the next entropy value for all the OD-flows is accomplished by multiplying the current estimated values $\hat{F}(t)$ by the time update matrix C obtained in the calibration phase. In this matrix the diagonal elements capture the time evolution of each OD-Flows whereas the non-diagonal elements capture the correlation between the OD-Flows. The predicted values $\tilde{F}(t)$ are derived from the time update equations :

$$\tilde{F}(t + 1) = C\hat{F}(t) \tag{2}$$

Estimation Step The estimated value for the next bin is defined as the predicted value adjusted by a correction factor:

$$\hat{F}(t + 1) = \tilde{F}(t + 1) + K(F(t + 1) - \tilde{F}(t + 1)) \tag{3}$$

where K is the Kalman gain matrix, which accounts for the confidence in the prediction model. The matrix K and the variances of the time series are updated between each iteration of the filter. The time series of the difference between the predicted and the estimated value $\eta(t) = \tilde{F}(t) - \hat{F}(t)$ represents the modeling error, and is often called the *innovation* in the Kalman literature.

Detection step If we assume that the model is correct, a large modeling error indicates an unexpected change in the associated IP header feature time series. Detecting anomalies consists of isolating these unexpected changes. In this paper we use the instantaneous method presented in [6]: an anomaly is detected on the i^{th} OD-flow at time t whenever $|\eta_i(t)| > T_h * \sigma_i$ where σ_i is the estimate of the variance of the i^{th} OD-Flow, η_i is the innovation of the i^{th} OD-Flow and T_h is the detection threshold.

3.5 Anomaly Database System

The original one week of IP flow traces contains roughly 100 gigabytes of data across both networks, and these traces are altered thousands of different ways according to our parameter space. In order to efficiently process such large amounts of data, the software that implements this part of our methodology was designed as a parallelized program written largely in C using LAM/MPI [18, 19] and run on a computer cluster. The set of all entropy timeseries is stored in a database along with the detected statistical outliers. Efficient access is important because the database is linked in with our Web-based front-end for classifying anomalies, which will be described in section 4.4. The MySQL database totals slightly over 30GB for both data and indexes.

4. ANOMALY CLASSIFICATION

In this section, we draw an important distinction between a statistical outlier (output by the detection technique) and a traffic anomaly (of interest to network operators). Then, we discuss our human-assisted methodology for classifying anomalies in terms of their start/end times. Our definition of anomalous regions also allows us to automatically calculate its type, which we define here. Finally, we briefly summarize our Web-based tool for classifying anomalies and generating the labeled traces we analyze in the rest of the paper.

4.1 Statistical vs. Traffic Anomalies

Statistical anomalies are the outliers detected by a given statistical-analysis technique. Statistical anomalies are therefore the output of the vast majority of traffic anomaly detectors, since they often utilize such techniques. Statistical anomalies are not what network operators are interested in, however; rather, they want to detect and diagnose the network events that affect their networks, such as DDoS attacks, worms, or port scans, which we define as the *traffic anomalies.* In an ideal world, therefore, we would evaluate anomaly detectors according to their ability to detect a wide variety of traffic anomalies.

Unfortunately, the root-cause analysis necessary to determine the traffic anomaly (potentially) associated with a given statistical anomaly is both too time consuming to be done manually and presently beyond automated techniques. It is simply not feasible to manually label entropy timeseries with millions of data-points, built from IP flow traces with even more flows, and altered thousands of different ways according to our parameter space. Yet, we still want a meaningful measure of ground truth that allows us to independently evaluate our traffic anomaly detector using metrics such as *false-positive* rate and *anomaly type* as a function of our parameter space. We therefore require a compromise definition of anomaly that that lies in between statistical anomalies and traffic anomalies.

4.2 Manual Labeling

We therefore define an *anomalous region* to be a sequence of statistical anomalies, which do not appear to be mere noise. We have found that this definition gives us deep insights into the impact of our parameter space on traffic anomaly detection, while also being efficient for a human operator to identify. Efficiency is important because the statistical anomalies in our one week of measurement data must be inspected thousands of times according to our parameter space, since every parameter alters the data itself.

While statistical-analysis techniques are admittedly much better at finding statistical outliers than humans, humans here provide a method-independent metric by which to evaluate a traffic anomaly detector. Moreover, statistical techniques do have flaws and shortcomings—e.g. potentially small threshold or a polluted definition of normal traffic patterns—which lead them to make mistakes that a human operator can detect. Our architecture is also built to allow multiple operators to classify the same statistical anomaly, which will allow us to achieve even greater confidence in our labeling.

As anomalous regions—and hence all the statistical anomalies within them—are defined to be true positives, the false-positive rate follows immediately. We will always specify it as a percent of all statistical anomalies. We do not provide any false-negative rate, however, because providing such labeled traces does not scale whatsoever. That is, the original one week of IP flow traces for two networks leads to timeseries with over 200 million data-points due to our parameter space, which clearly cannot be individually inspected by hand in order to identify *all* anomalous regions. It would, in fact, take one person over 6 years to manually label all our traces if he averaged one data-point inspected per second.

4.3 Anomaly Type

In addition to providing a meaningful false-positive rate, the identification of the anomalous region allows us to characterize the anomaly in terms of its statistical properties, e.g., how it alters the entropy timeseries. For example, the fact that an anomaly is a decrease in source and destination IP address entropy is strong circumstantial evidence for the hypothesis that the anomaly occurs between a small set of IP addresses. We have therefore constructed a definition of anomaly type that describes how the timeseries is changed by the anomaly, and can be determined automatically. That is, after the human labeler has identified the anomalous region, our tool will automatically determine and store its anomaly type.

Σ	meaning
$+$	$[+5\%, \infty\rangle$
$-$	$\langle\infty, -5\%]$
0	$\langle -5\%, +5\%\rangle$

Table 2: Meaning of alphabet for anomaly types

τ	Potential Description
−−−−−−	temporary outage
−−−−+−	DoS
−−−−++	alpha flow
−−−++−	port scan
−+−−+−	host scan
+−+−++	flash crowd
−+−+++	point-to-multipoint

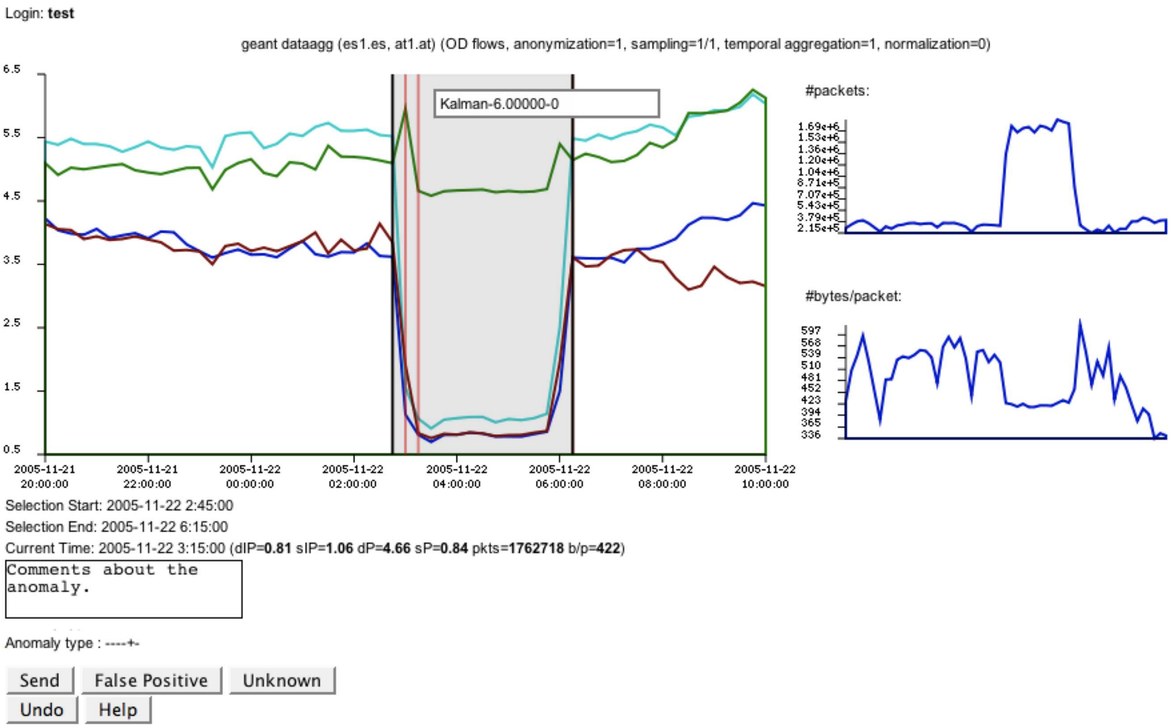
Table 3: Prevalent Anomaly Types

While only entropy timeseries are analyzed by our Kalman filter for detection, we include the change in two additional features in our anomaly type definition, namely the number of packets and bytes-per-packet (b/p) for a given OD-flow. These two features allow us, for example, to separate a likely alpha flow (an IP flow containing a large number of bytes and packets) from a possible DoS attack, both of which would correspond to decreases in source and destination IP address entropy timeseries, but have opposite effects on the (b/p) timeseries.

We define the start of the anomalous region as the last point *before* the statistical anomalies, whereas the end of the region is the first point in time *after* the statistical anomalies. The percent change between the mean of the endpoints and mean of values in-between determines whether a given feature s_i increased $(+)$, decreased $(-)$, or remained unchanged (0), as described in table 2. The type τ of an anomalous region (henceforth referred to as 'anomaly type' for convenience) is therefore defined as a sequence of six letters which describe how the six features mentioned above change during the anomalous period:

$$\tau \in \{`s_1 s_2 s_3 s_4 s_5 s_6\text{'} | s_i \in \Sigma\} \cup \{`xxxxxx\text{'}\} \qquad (4)$$

s_1 through s_6 refer to entropy timeseries for source IP addresses, destination IP addresses, source port num-



Login: **test**

geant dataagg (es1.es, at1.at) (OD flows, anonymization=1, sampling=1/1, temporal aggregation=1, normalization=0)

Selection Start: 2005-11-22 2:45:00

Selection End: 2005-11-22 6:15:00

Current Time: 2005-11-22 3:15:00 (dIP=**0.81** sIP=**1.06** dP=**4.66** sP=**0.84** pkts=**1762718** b/p=**422**)

Comments about the anomaly.

Anomaly type : ----+-

Figure 1: WebClass: Web-based Anomaly Classification Tool

bers, and destination port numbers, in addition to the packet and (b/p) timeseries. Table 3 gives descriptions for some of the more prevalent anomaly types. For example, the alpha flow we discussed earlier could have an anomaly type τ = ----++ because it also coincides with a concentration in the distribution of port numbers used and an increase in the total number of packets. The anomaly type 'xxxxxx' is to accommodate anomalies that do not have a discernible end, which prevents us from performing the endpoint versus middle comparison. These anomalies are most often sudden, persistent shifts in traffic. Furthermore, figures in the result sections that use our definition of anomaly type will also contain the distinction "other", which is not a separate anomaly type but rather the sum (in terms of frequency) of all anomaly types that are not individually distinguished in the graphs. Previous work has done clustering based on entropy values of the four IP header fields [7], but using the change in these fields to characterize an anomaly's type is a contribution of this paper.

4.4 Anomaly Classification Tool

WebClass is our web-based classifier that allows operators to label the start and end of anomalous regions and distinguish these from false positives; its main interface can be seen in figure 1. For every statistical anomaly, WebClass shows the associated OD-flow timeseries, which is parsed from the MySQL database described in section 3.4. Every anomaly is denoted by a red error line and upon mouse-over on any anomaly, the techniques and specific configurations that detected it are shown, as illustrated in the figure. The tool is built to allow multiple operators to label anomalies concurrently, which has allowed us to classify nearly 400 thousand statistical anomalies across over 20 thousand anomalous regions. We plan to make our tool available to the community, as we think it can play a valuable role in enabling reproducible research on anomaly detection using a larger collection of traces [1].

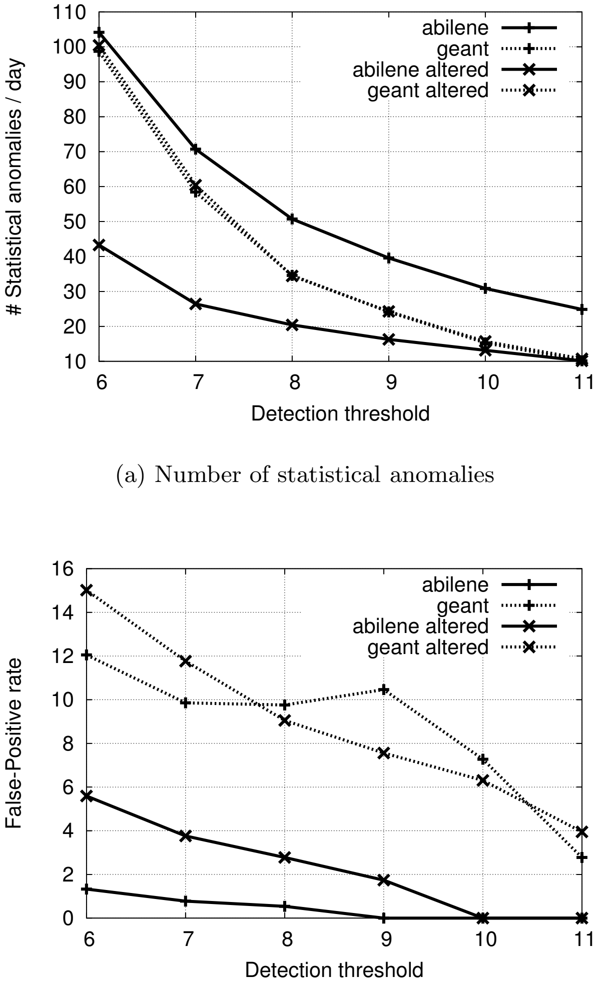
5. DEFAULT NETWORK PROPERTIES

The following section compares Abilene and Geant under (1) their default data-reduction configurations, and (2) the minimal amount of additional reduction necessary to put them in the same configuration. The former allows us to compare them in an unaltered setting with the most detailed data whereas the latter compares them on the same footing. In addition to highlighting the similarities and differences, we use the metrics we have previously defined to investigate the impact of altering the detection threshold. The differences observed between the two networks are explained by inherent properties of the two networks.

5.1 Anomaly Frequency

Figure 2(a) plots the number of statistical anomalies detected by our Kalman filter as a function of the detection threshold for both Abilene and Geant. There are

[1] A manual has already been written but cannot presently be shared due to the double-blind submission policy. A URL to access WebClass also already exists.



(a) Number of statistical anomalies

(b) False-positive rate

Figure 2: Impact of altering detection threshold

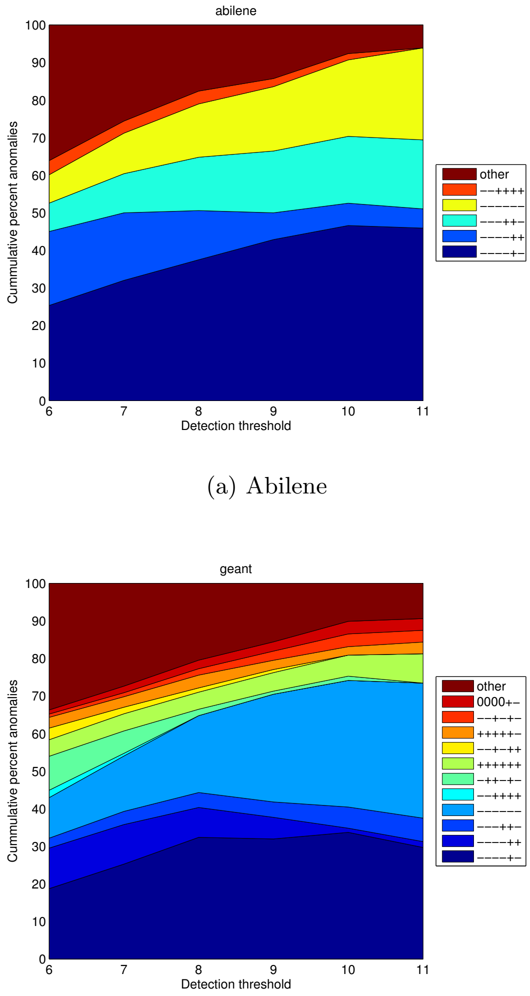
two curves for each network in order to allow us to evaluate the networks under (1) their default level of data-reduction, and also (2) under the same level of data-reduction (10^{-3} packet sampling, 11 bits IP anonymization, and 15-minute time bins). The curves using the default settings have the name of the respective networks whereas the "*network* altered" curves share the same data-reduction settings. Throughout our paper, all figures of this type are normalized to mean number of detections per day. Figure 2(b) plots the false-positive rate under the same conditions as figure 2(a). The former metric is important because it signifies the number of alarms that a network operator must deal with, and we have already established the importance of the false-positive rate. We chose not to explore threshold values lower than 6σ because we believe a mean rate of 100 anomalies per day is more than what a network operator will wish to deal with. Many small anomalies will also be included at such low detection thresholds; 6σ yields more than one statistical anomaly per time interval for Geant ($\frac{60}{15} \times 24 \times 7 = 672$ intervals total). It is possible to have more than one anomaly per time interval because every time interval contains multiple OD flows.

Figure 2(a) shows that while both networks have a similar number of statistical anomalies for low threshold levels under their default data-reduction levels (within 6% of one another for 6σ), there are much greater differences at higher levels (nearly 60% at 11σ). In other words, the drop off in number of anomalies as the detection threshold is increased is more pronounced for Geant (90%) than it is for Abilene (76%). While further IP anonymization barely affects that number of detected statistical anomalies for Geant, the added temporal aggregation and sampling greatly reduces the number of detected anomalies for Abilene at the lower detection thresholds. The impact of each of these data-reduction techniques will be studied, and explained, in further detail in later result sections.

Figure 2(b) shows a steady decline in false-positive rate for the Geant network between thresholds 6σ (12%) and 11σ (3%) for the default data-reduction levels, which is quite intuitive. The false-positive rate for the Abilene network, however, starts at a remarkably low 2% for detection threshold 6σ and reaches 0% already at detection threshold 9σ. The false-positive rates for the two networks are more similar when comparing them under the same data-reduction techniques, but Geant still has more than twice has many false positives, percentage wise. Note that while the overall trend of false-positive rate a function of the threshold is decreasing, this need not be a monotonic trend, as evidenced between 7σ and 9σ for Geant in figure 2(b). The reason that this phenomenon is not inconsistent is that it is possible to remove "false" anomalies at a faster rate than "true" anomalies, as the detection threshold is increased.

When taken together, figures 2(a) and 2(b) support the hypothesis that Abilene's traffic timeseries is more volatile than Geant's. That is, the definition of normal for the Abilene timeseries is already so volatile that any statistical anomaly that stands out even 6σ compared to this mean is quite significant. Geant, on the other hand, is more stable, which leads to a reasonable false-positive rate. While the two networks have more similar false-positive rates under the same data-reduction parameters, there is still a 2X difference and Abilene then has much fewer statistical anomalies. Our hypothesis that Abilene is more volatile than Geant is consistent with the stated purpose of the two networks. That is, not only is Abilene used for research traffic, but it does not provide access to the commodity Internet. For example, large individual bulk transfers between individuals correspond to a large percentage of anomalies in Abilene whereas such events are more frequently drowned out in Geant. We will further support this claim in the next subsection.

5.2 Anomaly Type Distribution



(a) Abilene

(b) Geant

Figure 3: Anomaly type distribution as a function of detection threshold for default data-reduction settings

The area plot in figure 3(a) shows the distribution of anomaly types as a function of the detection threshold for Abilene, and figure 3(b) is the same graph for Geant. These figures use the default data-reduction settings, for maximum information. All included anomalies have been labeled as true positives. Any anomaly type that corresponds to more than 5% of the total number of anomalies for any detection threshold is represented by a separate region, and the sum of all remaining anomaly types is included in the catchall group "other". All regions are listed in the legend in the same order as they appear in the figure. The graphs clearly illustrate that there is a great deal of homogeneity in Abilene anomalies, in that a very small number of anomaly types account for majority of detected statistical anomalies. Furthermore, this trend becomes more pronounced as the detection threshold is increased, to the point that only three anomaly types account for nearly 90% of statistical anomalies at 11σ, compared to more than twice the number of types for Geant at the equivalent threshold.

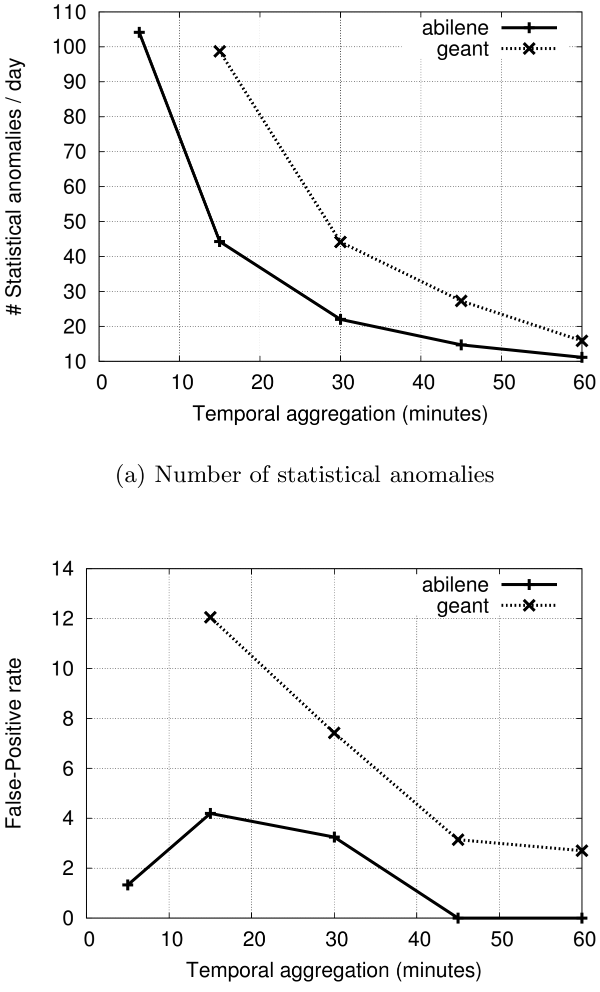
The networks do agree in the types of anomalies that are least sensitive to increases in the detection threshold, namely likely temporary outages ($\tau =$ ------) and potential DoS attacks ($\tau =$ ----+-), with the latter one more generally being a network event that increases the number of packets between two hosts using specific port numbers but very few bytes per packet. The graphs clearly show that high detection thresholds lead to a great loss of richness in the types of anomalies that are detected. In other words, determining how high the detection threshold should be set is not only a tradeoff between number of detected statistical anomalies and false-positive rate, but also a tradeoff between variety in anomalies and false-positive rate. For this reason, we will use detection threshold 6σ for all remaining plots in the paper unless we specify otherwise.

Finally, as was noted in the previous subsection, network events that involve only two hosts correspond to a much higher percentage of detected statistical anomalies in Abilene than Geant. Alpha flows ($\tau =$ ----++), for example, are detected about twice as frequently in the Abilene network compared to Geant. And potential port scans ($\tau =$ ---++-) are a prominent anomaly type in Abilene with over 10% of detected statistical anomalies at the higher detection thresholds whereas they are much less conspicuous in Geant. Again, this need not mean that there are more underlying port scans in Abilene than Geant, but rather that they manage to stand out against the less diverse background traffic in Abilene.

6. TEMPORAL AGGREGATION

The length of time bin chosen when discretizing IP flow traces determines the level of *temporal aggregation*, and has a strong impact on the ability to detect certain kinds of anomalies. We tested time bin lengths of 15, 30, 45, and 60 minutes for both networks, in addition to 5 minutes for Abilene. Lower values than these are not possible due to the networks' default configurations.

Figure 4(a) plots the number of statistical anomalies as a function of temporal aggregation, whereas figure 4(b) plots the impact of temporal aggregation on



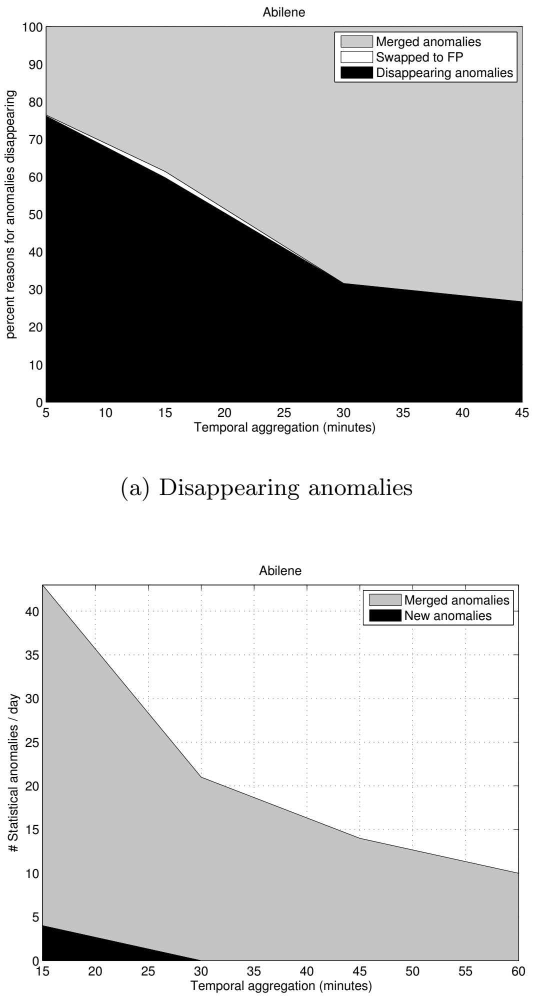
(a) Number of statistical anomalies

(b) False-positive rate

Figure 4: Impact of temporal aggregation

the false-positive rate. Figure 4(a) shows that the number of statistical anomalies decreases monotonically for both networks as temporal aggregation is increased. For Abilene, the percentage decrease is roughly proportional to the decrease in number of points in the timeseries whereas for Geant the rate of decrease is slightly higher.

As explained in section 5.1, Abilene has a very low false-positive rate to begin with, and hence additional temporal aggregation does not significantly impact its false-positive rate, as seen in figure 4(b). Geant's false-positive rate monotonically decreases as a function of the temporal aggregation. An improved false-positive rate is a natural consequence here since only the largest anomalies will remain significant enough to be detected despite the longer time window (and hence other traffic) within which they are placed. This improvement is insignificant, however, when compared with the loss of in number of statistical anomalies seen in figure 4(a).



(a) Disappearing anomalies

(b) Appearing anomalies

Figure 5: Reasons for anomalies disappearing and appearing

There are several possible reasons for why the number of statistical anomalies decreases in figure 4(a) as a function of the temporal aggregation. Two adjacent statistical anomalies may be *merged* into one detection at a higher level of temporal aggregation. The other primary alternative is that a smaller traffic anomaly becomes statistically insignificant when placed in a larger time bin, and hence *disappears*, which means that the underlying network event goes unnoticed. Finally, our methodology introduces a human element, which might introduce error or ambiguity as an anomaly is *swapped* from true-positive to false-positive when increasing the temporal aggregation.

The percent contribution of each of the above reasons for the decrease in number of statistical anomalies can be seen in figure 5(a), which is plot for Abilene. The graph shows that, for lower aggregation levels t_i, the decrease in number of statistical anomalies when going to t_{i+1} is predominantly due to anomalous regions simply disappearing. This supports the conclusion that shorter-lived anomalies are "lost" early as one aggregates further. As the aggregation increases, however, the decrease in number of statistical anomalies between t_i and t_{i+1} is increasingly caused by anomalies being merged. Our experience has been that only very large and relatively long-lived anomalies remain at this point, which are combined into one time bin upon further temporal aggregation.

Naturally temporal aggregation is not exclusively a negative thing. Longer time bins lead to sparser time-series, hence less data being stored and processed by the traffic anomaly detector. Moreover, some amount of temporal aggregation can reveal important temporal correlations in the underlying trace. That is, increased temporal aggregation may reveal traffic anomalies that straddle multiple time bins at lower levels but may not be conspicuous enough in any individual time bin to be classified as an outlier. In order to investigate the frequency of this phenomenon, figure 5(b) plots for any given temporal aggregation level t_i, the proportion of statistical anomalies that are the result of a *merging* of statistical-anomalies that were already detected at t_{i-1} versus entirely *new* anomalies relative to t_{i-1}. In other words, figure 5(b) explains the area under the curve of figure 4(a) for Abilene. The figure clearly shows that time bins larger than 5 minutes reveal very few new anomalies. That is, less than 10% of all anomalies detected when using a 15-minute time window were not already detected using the 5-minute window, and a 30-minute window gains nothing relative to 15 minutes. When taken together with figure 5(a), this strongly discourages using time windows longer than 5 minutes.

Figure 6 shows the distribution of anomaly types as a function of the temporal aggregation for Abilene. The graph shows that increasing the temporal aggregation significantly reduces the diversity of anomalies. Taken together with figure 5(a), it shows that the anomalies that get drowned out with an increase in aggregation level are not only short-lived but also frequently the rarer ones. In other words, the most prevalent anomalies, e.g. temporary outages ($\tau =$ ------) and potential DoS attacks ($\tau =$ ----+-), are most impervious to temporal aggregation. Our experience has been that the very long-lived anomalies are often temporary outages, which were already easy to detect at lower levels of temporal aggregation.

Figures 7(a) and 7(b) show the impact of sweeping both the temporal aggregation and detection threshold on the number of detected statistical anomalies and false-positive rate, respectively. The graphs show that the interaction between these two knobs is quite intuitive: both the number of statistical anomalies and the false-positive rate decrease nearly monotonically as a function of the temporal aggregation, regardless of the detection threshold—i.e. at no point do the curves intersect. The graphs also highlight the importance of using multiple metrics when determining how to set the level of temporal aggregation and detection threshold. That is, aggregating into 60-minute time bins will lead to a false-positive rate of 0% even when using a detection threshold as low as 7σ (similarly for any choice of time bin greater than or equal to 30 minutes when using detection threshold greater than or equal to 9σ), but this comes at a great cost in terms of number of statistical anomalies and variety of anomalies that can be detected. While these two figures relate only to Geant, the Abilene plots are quite similar.

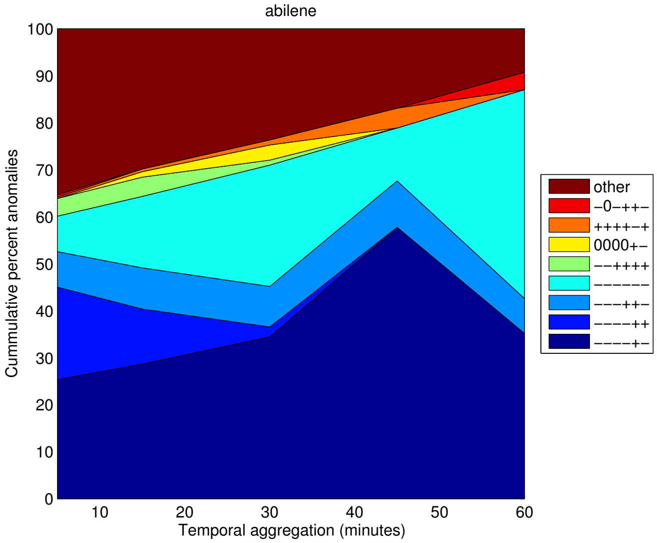
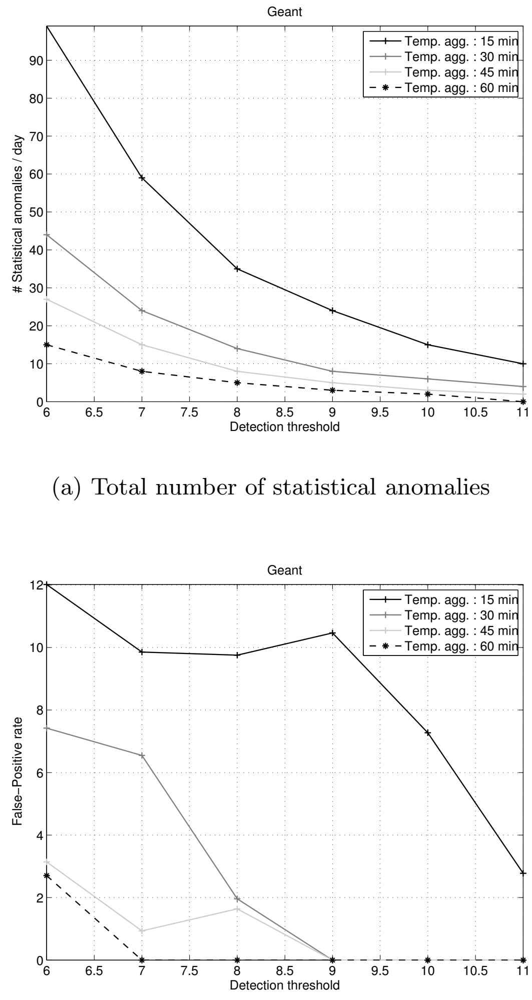


Figure 6: Anomaly distribution of Abilene for temporal aggregation

In summary, temporal aggregation beyond 5 minutes is not advisable. Even using such "short" time intervals (i.e. shorter than the default for Geant), the benefit of the few numbers of new anomalies detected by increased aggregation is greatly outweighed by the loss of smaller and less frequent anomalies. Moreover, from a practical standpoint, longer time bins straightforwardly mean that it takes more time to "fill" a bin, which worsens response time in an online detection and diagnosis setting. Fortunately, the detection threshold and level of temporal aggregation interact intuitively, which makes it easier to predict the outcome of tuning them.

7. IP ADDRESS ANONYMIZATION

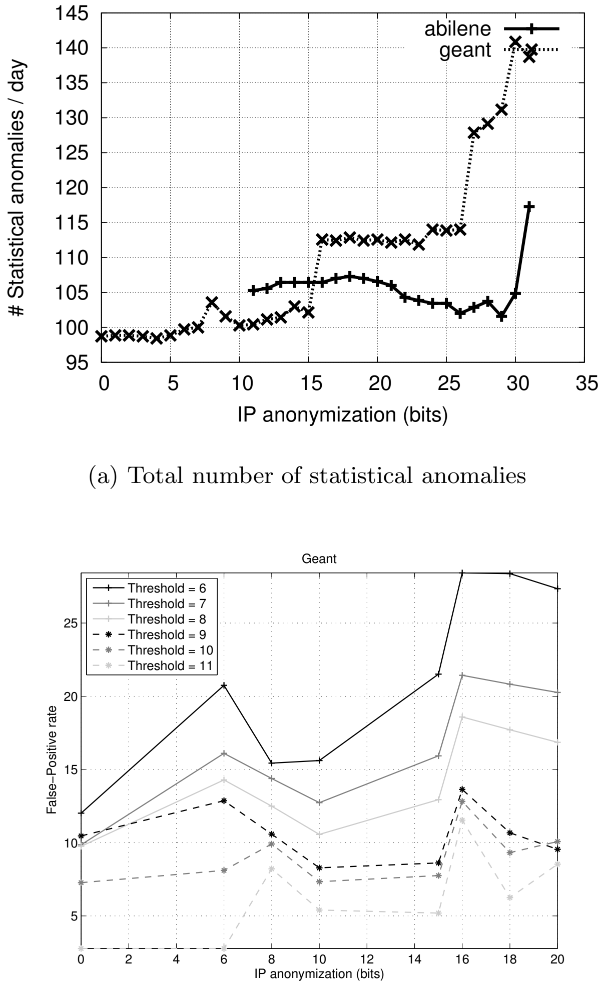
The number of bits anonymized from IP addresses is the *IP anonymization* parameter. We evaluated anonymiza-



(a) Total number of statistical anomalies

(b) False-positive rate

Figure 7: Impact of temporal aggregation on Geant across multiple detection thresholds



(a) Total number of statistical anomalies

(b) False-positive rate

Figure 8: Impact of IP anonymization

tion values of 0 through 31 bits, inclusive, but will only analyze false-positive rates for Geant due to space constraints.

Figure 8(a) plots the number of detected statistical anomalies as a function of the IP anonymization for both Abilene and Geant. Note that Abilene begins at 11 bits of anonymization because this is the level at which its IP traces are exported. Furthermore, overly aggressive anonymization becomes semantically meaningless; because every bit anonymized reduces the space of IP subnets by half, the increases at aggregations higher than /5 correspond to only 32 subnets, which results in very small entropy values and a substantial amounts of noise. This can be further observed in figure 8(b) for Geant, which plots the false-positive rate as a function of IP anonymization, and shows a strongly increasing trend to the point that nearly 30% of detected anomalies are false alarms when anonymizing 16 bits.

While IP anonymization tends to increase the overall false-positive rate, it may make certain kinds of anomalies easier to detect. That is, an IP scan corresponds to a dispersion in the distribution of destination IP addresses seen, but this dispersion may not be significant enough to cause a detection without further aggregation. But when IP addresses are anonymized, the consecutive addresses scanned may become part of a single IP subnet and hence be detected as a concentration. This can clearly be seen in figure 9, which plots a single anomaly that occurred on the 21st of December, 2005 and traversed the Geant network from Greece (gr1) to Germany (de2). Each line corresponds to a different level of IP anonymization, with the upper lines being less aggregated (e.g., /32) whereas the lower lines are

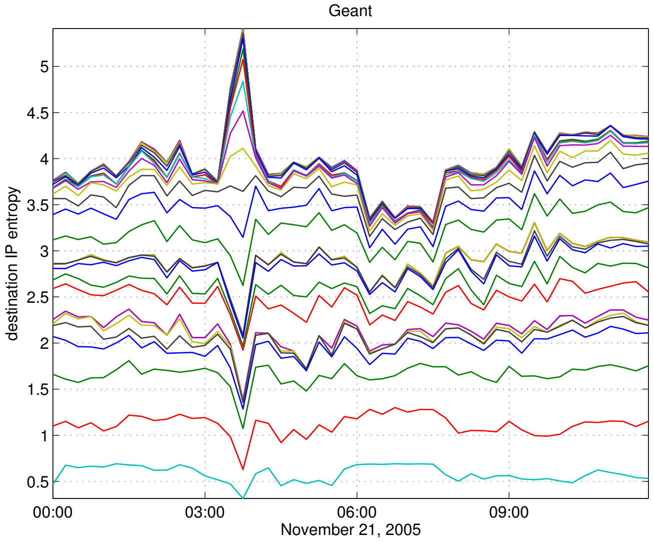


Figure 9: IP scan transformed from dispersion to concentration

heavily aggregated. The striking feature of these plots is that the same underlying network event causes an increase in entropy at lower aggregation levels and a decrease at higher aggregation levels.

Figure 9 also makes a strong argument for leveraging IP address aggregation for traffic anomaly detection and diagnosis. For example, concurrently analyzing multiple IP anonymization levels would allow a traffic anomaly detector to both detect and diagnose IP scans by searching for the symptom observed in this figure. In addition, it is likely that the exact aggregation levels at which the positive and (separately) negative change in entropy is the greatest, would give insights into the scope (in terms of number of affected IP addresses) and potentially location, of the IP scan.

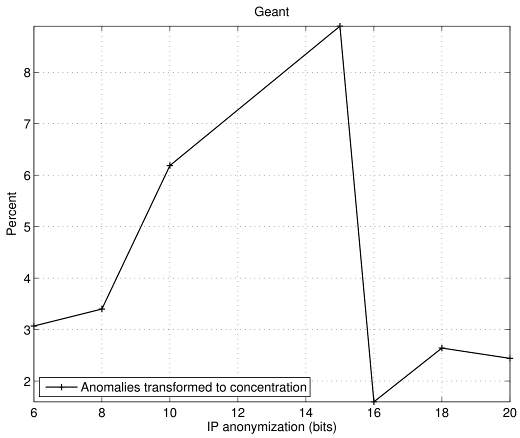


Figure 10: Percent anomalies transformed into IP address space concentrations

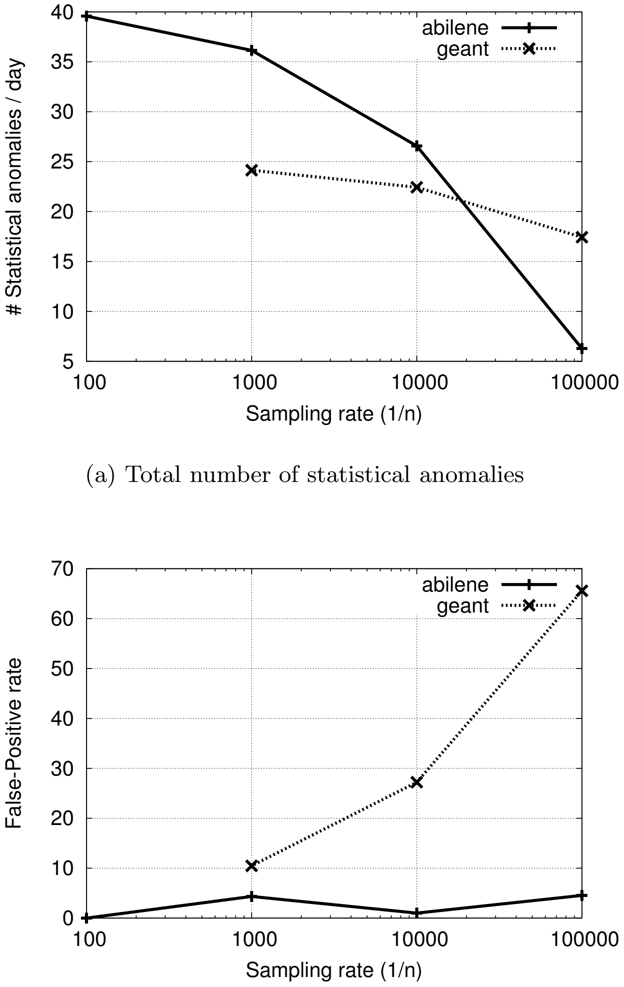
In order to investigate the prevalence of dispersions being transformed into concentrations by anonymizing IP address bits, figure 10 plots the percent of anomalous regions that are swapped from a '+' or '0' (corresponding to a potential dispersion) to a '-' (a concentration) as we increase the IP anonymization. The plot shows that there is a marked increase in the percentage of anomalies that are transformed into concentrations between 8 and 14 bits. IP address anonymization has admittedly nontrivial interactions with the entropy timeseries, which makes it a difficult data-reduction parameter to analyze, but this figure suggests that a significant fraction of detected IP scans correspond to /8 to /15, which corresponds to between 256 and 32k hosts scanned.

8. SAMPLING

The large volume of traffic that passes through modern backbones forces network operators to perform packet sampling. Router manufacturers may implement different sampling techniques, such as "pick every n'th packet" or "randomly pick 1 in n packets", but we chose to use only the latter policy to further sample our data sets. The *sampling rate* chosen at a given network is very important and we have found it to have a strong impact on the ability to detect anomalies. Abilene and Geant have different default sampling rates, which means that 10^{-2} could only be evaluated for Abilene, but $10^{-3}, 10^{-4}, 10^{-5}$ were studied for both networks. Note that the plots in this section are the only ones in our paper where the detection threshold is set to a value (9σ) other than 6σ. The reasons for the change will be explained later in this section.

Figure 11(a) plots the number of statistical anomalies as a function of the sampling rate for both Geant and Abilene. The curves for both networks have a downward trend, with an acceleration in fall off as sampling is increased. Not only does sampling lead to fewer detected statistical anomalies, but we have also found that aggressive sampling leads to a tremendous loss in detail, as can be seen in figure 11(b), which plots the false-positive rate as a function of the sampling rate. Both networks show an increase in the percent of false positives as sampling is increased, but clearly the increase for Geant dwarfs that for Abilene. We believe the reasons for this discrepancy is largely explained by the differences between the two networks—detailed in section 5.1—primarily the volatility in Abilene's trace, which leads to a lot of "true" anomalous regions.

The reason that plots in this section are not using the default 6σ detection threshold is because the interaction between detection threshold and sampling rate is more complicated than for the other data-reduction techniques. Figure 12(a) plots the number of statistical anomalies as a function of the detection threshold



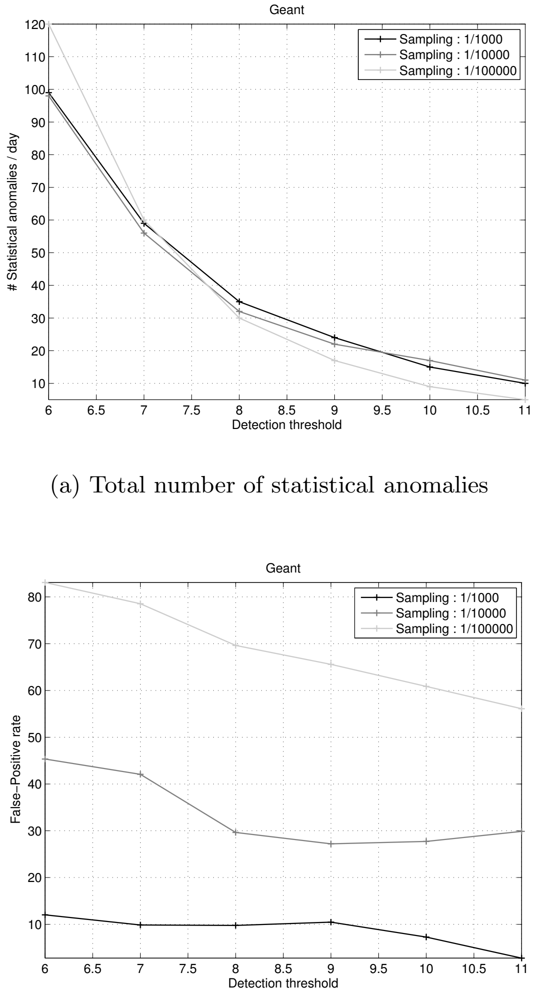
(a) Total number of statistical anomalies

(b) False-positive rate

Figure 11: Impact of sampling

for multiple sampling rates. Notice how the curves intersect, i.e. the highest sampling rate has the *highest* number of detections at the lowest detection threshold but the *lowest* number of detections at the highest detection threshold. The reason for this is that sampling tends to make timeseries increasingly spiky. At very low detection thresholds, all of these spikes are marked as being outliers whereas at high detection thresholds these spikes are more correctly interpreted as noise. The increased level of noise leads to extremely high false-positive rates—seen in figure 12(b)—due to the poor diversity in traffic in conjunction with small volumes caused by high sampling rates, which make statistical techniques ineffective.

Independent of which network is studied, it is very clear that overly aggressive sampling can have a profound negative impact on one's ability to detect traffic anomalies. The impact is not only measured in the false-positive rate, but also in the types of anomalies that can be detected. That is, it has long been assumed that sampling rate has a more detrimental impact on network events that involve fewer packets, and figure 13 provides tangible support for this hypothesis. The figure plots the anomaly type distribution as a function of the sampling rate for Abilene, and there is a clear loss of variety. While the group "other" corresponds to over 30% of the anomalies at 10^{-2}, this number is monotonically reduced to nearly 5% when sampling at 10^{-5}. Once again we see that temporary outages ($\tau =$ `------`) and potential DoS attacks ($\tau =$ `----+-`)—both of which can drastically alter the number of packets in a given interval—are least affected by data-reduction techniques. While a higher sampling



(a) Total number of statistical anomalies

(b) False-positive rate

Figure 12: Impact of sampling on Geant for multiple detection thresholds

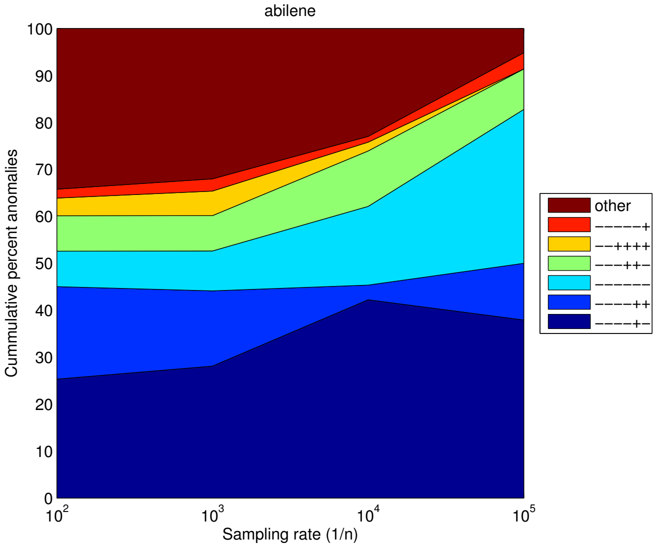


Figure 13: Anomaly distribution for sampling

rate reduces the CPU load on routers, the loss in both number and variety of detected anomalies is also compounded by a very large increase in the false-positive rate. For these reasons, the sampling rate is a knob that must be tuned with great care—the lower you are able to keep the sampling rate, the better traffic anomaly detectors will tend to perform.

9. CONCLUSIONS

Data-reduction techniques are an unavoidable part of anomaly detection in modern IP networks, both to reduce overhead and to reveal underlying correlations. In this paper, we have studied the effects of three main data-reduction techniques—temporal aggregation, IP address aggregation, and sampling—on the effectiveness of anomaly detection on traffic-matrix data in two backbone networks. In some cases, the data-reduction techniques make it *easier* to detect anomalies. For example, IP address anonymization can make IP scans easier to detect and diagnose, and temporal aggregation may reveal some anomalies that straddle multiple time bins at smaller levels of aggregation. However, for the most part, data-reduction techniques reduce the effectiveness of anomaly detection by lowering the number of detectable anomalies. In particular, excessive sampling and aggregation dilute most anomalies except for very large changes in traffic volume. Sampling, in particular, significantly increases the false-positive rate by making the entire timeseries more spiky.

Our results suggest a number of promising avenues for future research. We explored only part of a large parameter space. For example, future work could go beyond OD flows to consider other network-wide representations of the traffic, such as the link matrix. Another natural direction to pursue is evaluating, and comparing, other statistical-analysis techniques. We believe that our evaluation methodology, and our software tools, would be quite useful for these future studies. In addition, the value of our anomaly-classification system will increase as more researchers use it; for example, with more people independently classifying each anomaly, we could provide confidence intervals for the labeling of the data. Further manual exploration of the data could shed light on the underlying causes of the anomalies we detect, which could help refine our methodology for automatically identifying the anomaly type. Finally, the many differences we see between the Abilene and Geant data suggest that it would be valuable to analyze more traces, from more networks, to investigate whether our conclusions can be further generalized.

In summary, our research shows that data-reduction techniques have a significant impact on the effectiveness of traffic anomaly detection. Our primary contributions are therefore: (1) a methodology and tool for sweeping the large associated parameter space, (2) a methodology and tool for evaluating the effectiveness of traffic anomaly detectors based on the false-positive rate and a novel definition of anomaly type, and (3) a first step towards quantifying the effects of data-reduction techniques to allow operators to make informed trade-offs between the number/type of anomalies they wish to detect and the system overheads they must endure.

10. REFERENCES

[1] C. Estan, S. Savage, and G. Varghese, “Automatically inferring patterns of resource consumption in network traffic,” in *ACM SIGCOMM*, (Karlsruhe, Germany), pp. 137–148, 2003.

[2] Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund, “Online identification of hierarchical heavy hitters: Algorithms, evaluation, and applications,” in *ACM Internet Measurement Conference*, (Taormina, Sicily, Italy), pp. 101–114, 2004.

[3] P. Barford, J. Kline, D. Plonka, and A. Ron, “A signal analysis of network traffic anomalies,” in *ACM Internet Measurement Workshop*, (Marseille, France), pp. 71–82, 2002.

[4] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, “Sketch-based change detection: Methods, evaluation, and applications,” in *ACM Internet Measurement Conference*, (Miami Beach, FL, USA), pp. 234–247, 2003.

[5] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, “Network anomography,” in *ACM Internet Measurement Conference*, (Berkeley, California, USA), October 2005.

[6] A. Soule, K. Salamatian, and N. Taft, “Combining filtering and statistical methods for anomaly detection,” in *ACM Internet Measurement Conference*, (Berkeley, California, USA), October 2005.

[7] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” in *ACM SIGCOMM*, (Philadelphia, Pennsylvania, USA), pp. 217–228, 2005.

[8] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in *ACM SIGCOMM*, (Portland, Oregon, USA), pp. 219–230, 2004.

[9] A. Soule, H. Ringberg, F. Silveira, J. Rexford, and C. Diot, “Detectability of traffic anomalies in two adjacent networks,” *Passive And Active Measurement Conference*,

2007.

[10] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is sampled data sufficient for anomaly detection?," in *ACM Internet measurement Conference*, (Rio de Janeriro, Brazil), pp. 165–176, 2006.

[11] J. Mai, A. Sridharan, C.-N. Chuah, H. Zang, and T. Ye, "Impact of packet sampling on portscan detection," *IEEE Journal on Selected Areas in Communication*, vol. 24, December 2006.

[12] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," in *ACM Internet measurement Conference*, (Rio de Janeriro, Brazil), pp. 159–164, 2006.

[13] Abilene Backbone Network. `abilene.internet2.edu/`.

[14] Geant Network. `www.geant.net/`.

[15] Juniper J-Flow. `www.juniper.net/techpubs/software/erx/junose61/swconfig-routing-vol1/ht%ml/ip-jflow-stats-config2.html`.

[16] Abilene Participation Agreement. `abilene.internet2.edu/community/connectors/AbileneConnectionAgreement20%06.pdf`.

[17] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Tech. Rep. CRG-TR-96-2, University of Toronto, February 1996.

[18] G. Burns, R. Daoud, and J. Vaigl, "LAM: An Open Cluster Environment for MPI," in *Proceedings of Supercomputing Symposium*, pp. 379–386, 1994.

[19] J. M. Squyres and A. Lumsdaine, "A Component Architecture for LAM/MPI," in *Proceedings, 10th European PVM/MPI Users' Group Meeting*, (Venice, Italy), 2003.