

Parametric Methods for Anomaly Detection in Aggregate Traffic

Gautam Thatte, *Student Member, IEEE*, Urbashi Mitra, *Fellow, IEEE*, and John Heidemann, *Senior Member, IEEE*

Abstract—This paper develops parametric methods to detect network anomalies using only aggregate traffic statistics, in contrast to other works requiring flow separation, even when the anomaly is a small fraction of the total traffic. By adopting simple statistical models for anomalous and background traffic in the time-domain, one can estimate model parameters in real-time, thus obviating the need for a long training phase or manual parameter tuning. The proposed bivariate Parametric Detection Mechanism (bPDM) uses a sequential probability ratio test, allowing for control over the false positive rate while examining the trade-off between detection time and the strength of an anomaly. Additionally, it uses both traffic-rate and packet-size statistics, yielding a bivariate model that eliminates most false positives. The method is analyzed using the bitrate SNR metric, which is shown to be an effective metric for anomaly detection. The performance of the bPDM is evaluated in three ways: first, synthetically-generated traffic provides for a controlled comparison of detection time as a function of the anomalous level of traffic. Second, the approach is shown to be able to detect controlled artificial attacks over the USC campus network in varying real traffic mixes. Third, the proposed algorithm achieves rapid detection of real denial-of-service attacks as determined by the replay of previously captured network traces. The method developed in this paper is able to detect all attacks in these scenarios in a few seconds or less.

Index Terms—Distributed denial of service (DDoS), anomaly detection, aggregate traffic, parametric models

I. INTRODUCTION

Security in computer networks is an extremely active and broad area of research, as networks of all sizes are targeted daily by attackers seeking to disrupt or disable network traffic. A successful denial-of-service attack degrades network performance, resulting in losses of several millions of dollars [14]. Development of methods to counter these and other threats is thus of high interest. Current countermeasures under development focus on detection of anomalies and intrusions, their prevention, or a combination of both.

In this paper, we present an anomaly detection method that profiles normal traffic; a traffic-rate shift and a change in the distribution of packet sizes from the nominal condition is flagged as an anomaly. Our anomaly detection problem is posed as a statistical hypothesis test. We develop parametric statistical models for typical and anomalous traffic. Our detection method does not need, or attempt, to model the full traffic patterns, instead it captures key, gross features of the traffic to enable informed decisions about changes in traffic.

We underscore that our model does *not* capture all aspects of general Internet traffic. However, we show that, in spite of this known mismatch, our model effectively captures changes in the traffic which are associated with network anomalies. Our goal is to see whether these simple, approximate statistical models can yield detection methods of high performance by modeling sufficient, salient features of the traffic.

Our approach has three key features. First, our model for anomaly detection *operates on aggregate traffic, without flow-separation or deep-packet inspection*. Both of these characteristics are essential for a practical and deployable anomaly detection system. Flow separation, per-flow anomaly detection, and deep-packet inspection are difficult or impossible for most backbone routers, which have tens to hundreds of thousands of active flows per minute [8]. Since our approach only considers packet headers and timing information, it is robust to traffic concealment via encryption or tunneling. While it is true that the source and destination IP addresses of each packet are always available at the routers, port numbers are not available without flow-separation. Some prior work [24], [21] uses features related to the source and destination port numbers and so will not be able to detect anomalies in aggregate or VPN tunneled traffic. Note that operating on aggregate traffic is sufficient to *detect* anomalies; we assume that responses such as filtering can involve heavier weight, per-flow analysis.

Second, unlike prior anomaly detection approaches, *our method automates training* and does not require hand-tuned or hard-coded parameters. Instead, key algorithmic parameters are automatically calculated based on the underlying model parameters, or estimates thereof, which evolve as a function of network traffic. For instance, the update window size, an algorithmic parameter which is described in Section IV, is computed based on the average sample number (ASN) function. The latter is a function of the underlying model parameters, and is derived in Appendix B. Our automation significantly eases deployment and operation in networks where traffic and anomalies inevitably evolve over time.

Third, we employ both the packet rate and the sample entropy of the packet-size distribution statistics to ensure robustness against false positives, thus overcoming one of the traditional drawbacks of anomaly detection methods. Combining both these features ensures that the detection of an anomaly is declared only when an increase in the traffic volume is accompanied by a change in the packet-size distribution. Thus, an increase in the background traffic alone will usually not be misidentified as an anomaly. We show that our proposed detection method successfully detects attacks with these features. Several real-world network phenomena have these features: all rate-based denial-of-service (DoS) attacks, including TCP

G. Thatte and U. Mitra are with the Ming Hseih Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089 USA. e-mail: {thatte,ubli}@usc.edu

J. Heidemann is with the Information Sciences Institute, University of Southern California, Marina Del Rey, CA, 90292 USA. e-mail: johnh@isi.edu

SYN attacks, ping attacks, and service request attacks that use fixed-sized requests and act upon DNS or web servers. Since an adversary may try to conceal fixed-size requests, we show that attacks from a *smart* attacker that attempts to vary request size (Section V-E) are also successfully detected by the bPDM. Finally, since the bPDM ignores packet addresses and contents, it can detect attacks with spoofed source addresses, and attacks in virtual private networks or with encrypted traffic in the common case that encryption does not systematically alter packet sizes.

The contribution of our paper is to develop the bivariate Parametric Detection Mechanism (bPDM), which is completely passive, incurs no additional network overhead, and operates on aggregate traffic. Furthermore, this work suggests it is feasible to detect anomalies and attacks based on aggregate traffic at network edges, and not just near attack victims. Our detection method employs the sequential probability ratio test (SPRT) [39], a time-adaptive detection technique, for the two aggregate traffic features we consider: packet rate and packet size. Combining the SPRTs for these two features ensures the bPDM is robust against false positives, yet maintains rapid detection capabilities. We validate the bPDM and quantify the method’s effectiveness on controlled synthetic traces, emulated Iperf attacks, and real network attacks. We introduce the bitrate SNR, which is found to be an effective metric for evaluation and superior to the previously proposed packet SNR metric [17]. Our algorithm also performs comparably to or better than a selected set of existing detection schemes, while mitigating key drawbacks via the features described above.

II. RELATED WORK

In this section, we review the prior art in anomaly and attack detection relevant to our work. The methods described can be broadly classified as techniques requiring flow-separation, spectral or frequency-domain methods, and non-parametric change-point methods.

Methods requiring flow-separation: The techniques in [13], [15], [19], [24], [25], [27], [29], [31], [38] and [40] use certain flow-separated traffic parameters, *e.g.* source and destination IP addresses and port numbers, to detect an attack. Flow-separated parameters are also employed for fast portscan detection [21], which uses an SPRT to develop an online detection algorithm.

These methods use header information and flow-separated features to detect anomalies and attacks, and in comparison to methods that classify outliers based only on traffic volume [33], [34], are more far more accurate while also yielding a lower probability of false positives. On the other hand, the main disadvantages of flow-separation are its inherent complexity at the router and its inability to process encrypted traffic. Our work operates on aggregate traffic, using the traffic volume (specifically, the packet rate) to detect attacks, with the improvement that incorporation of the entropy of the packet size, which does not require flow-separation, reduces the probability of false positives and allows us to discriminate between true attacks and non-malicious changes in traffic.

Non-parametric methods: This class of methods does not assume an underlying model, but rather tailors its detection

mechanism to the data. A variety of non-parametric methods employ CUSUM to implement change-point detection. The CUSUM algorithm [7] involves the calculation of a cumulative sum of the weighted observations. When this sum exceeds a certain threshold value, a change in value is declared. Prior work has focused on detecting SYN attacks using both aggregate traffic [33] and flow-separated traffic [40]. The work of [36] focuses on anomaly detection using features and statistics of the IP layer. Kalman filtering to detect anomalies using IP address filtered traffic is considered in [30]. A key drawback of the CUSUM algorithm is that the intensity of the anomaly needs to be known *a priori*; in most cases, the solution to this problem requires empirically-designed thresholds that necessitate significant human effort before the scheme is initially deployed. In contrast, our detection mechanism automatically calculates key algorithmic parameters based on the underlying model.

Spectral methods: Spectral techniques have been widely used in many other fields to distinguish hidden patterns and trends from a noisy background. In the past few years, researchers have begun to apply these methods to analyze network traffic. Spectrum-based approaches have been used to detect features with near-periodic signatures, such as bottlenecks in the link layer, the effects of the TCP windowing mechanism and DoS attacks [17], and traffic anomalies [6]. They have also been employed for attack fingerprinting [18]. The work in [9] used the energy spectrum to distinguish between reduction-of-quality flows and legitimate TCP flows in a distributed setting, and using the sequential SPRT framework. However, the detection accuracy of spectral methods degrades as the periodicities in the attack weaken, and most methods are more computationally expensive than corresponding time-domain techniques, especially when high speed aggregate traffic must be analyzed.

Our previous work [34] developed the parametric Modeled Attack Detector (MAD), which employed Poisson and shifted Poisson models that could rapidly detect low-rate attacks but required a dedicated training phase to learn the background traffic parameters, and which was susceptible to a few false positives. Furthermore, the one-parameter Poisson model did not allow for continuous updating of the background parameters, and suffered from overdispersion and underdispersion, given real network data. The bPDM discussed in this paper employs richer models that circumvent the need for a training phase. Combining the packet-rate and packet-size distributions nearly eliminates false positives. We present the bPDM in Section IV, but first provide an overview of sequential detection, which is the underlying framework of our anomaly detection method.

III. BACKGROUND IN SEQUENTIAL DETECTION METHODS

Hypothesis testing exploits prior knowledge of statistical descriptions of data in order to decide among a set of candidate populations [23]. In our problem setup, we have two hypotheses:

$$\begin{aligned} H_0 &: \text{No anomaly,} \\ \text{and } H_1 &: \text{Presence of an anomaly in traffic.} \end{aligned}$$

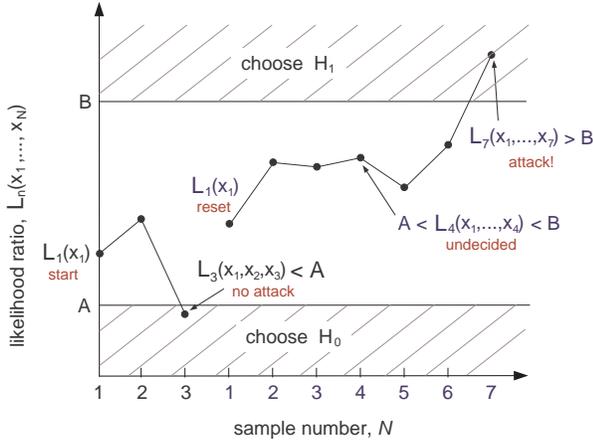


Fig. 1. Depiction of the sequential probability ratio test (SPRT).

The conditional probability density when hypothesis H_i is true is denoted $p(x|H_i)$ for $i = 0, 1$. We assume independent and identically distributed observations $\{x_k, k = 1, 2, \dots\}$ which are drawn from one of the two probability distributions.

Given the two hypotheses and thus two decision choices, there are four possible scenarios, of which we focus on two. A *false positive* (FP), or *false alarm*, is declared when the algorithm selects H_1 when H_0 is in fact true; choosing H_0 even though H_1 is true is termed a *false negative* (FN). The probabilities of these two scenarios,

$$\alpha = P_{FP} = \Pr[H_1|H_0] \quad \text{and} \quad \beta = P_{FN} = \Pr[H_0|H_1], \quad (1)$$

are used to specify the performance criterion of the sequential detection test. The bPDM employs the sequential probability ratio test (SPRT) [39] in order to quickly detect an attack.

The *likelihood ratio* is used to implement the SPRT. Given N independent and identically distributed observations $\mathbf{x} = \{x_1, \dots, x_N\}$, the likelihood ratio $L_N(\mathbf{x})$ is defined as

$$L_N(\mathbf{x}) = \prod_{k=1}^N \frac{p(x_k|H_1)}{p(x_k|H_0)} = \frac{p(x_k|H_1)}{p(x_k|H_0)} \cdot L_{N-1}(x_1, \dots, x_{N-1}), \quad (2)$$

where the second equality illustrates that the likelihood ratio can be easily updated given a new observation.

Given a new observation, the likelihood ratio is compared to two thresholds A and B which correspond to choosing H_0 or H_1 , respectively. Figure 1 depicts a realization of the SPRT wherein if $A < L_N(x_1, \dots, x_N) < B$, the sequential test continues, and an additional observation x_{N+1} is taken as is the case with L_4 in Figure 1. But if $L_N(x_1, \dots, x_N) \geq B$ or $L_N(x_1, \dots, x_N) \leq A$, then the test terminates and we choose hypothesis H_1 if the former, or hypothesis H_0 if the latter, is true. In Figure 1, we see that $L_3 < A$, and thus H_0 is chosen; then, the sequential test and likelihood ratio are reset since an anomaly was not detected, and the SPRT continues. When the likelihood ratio crosses either threshold, at say sample m , the sequential test is reset by computing the updated likelihood ratio as $L(x_{m+1})$ instead of $L(x_1, \dots, x_m, x_{m+1})$. We then see that $L_8 > B$, so H_1 is chosen, indicating that an anomaly has been detected. We can either stop the test now (as shown

in Figure 1), or reset the SPRT and see whether the likelihood ratio crosses threshold B again, potentially confirming the presence of an anomaly. This latter methodology is employed in the design of our detection mechanism, detailed in Section IV.

Ideally, the boundaries A and B are selected to minimize the probability of error for all possible values of N ; however this formulation of the problem is generally intractable and thus we use Wald's approximations [39] to approximate

$$B \cong (1 - \beta)/\alpha \quad \text{and} \quad A \cong \beta/(1 - \alpha), \quad (3)$$

which are functions of the required detection performance parameters from (1). We observe that the approximate values of A and B are independent of $p(x|H_i)$. The number of samples required for a particular test to make a decision is a random number. Thus, we examine the average value of this random number, referred to as the average sample number (ASN) function, to measure the efficacy of the SPRT. For the binary hypothesis test, the ASN function is denoted $\mathbb{E}_i(N)$ for hypothesis H_i , and is derived for our models in Appendix B.

IV. THE PARAMETRIC MODEL

In this section, we derive the SPRTs for the packet-rate and packet-size features that are the primary components of the bivariate Parametric Detection Mechanism (bPDM). The bPDM operates on a unidirectional sampled time-series of aggregate network traffic. The parametric models employed to derive the bPDM are *not* representative of *general* Internet traffic, but rather are chosen to differentiate between the presence-of-anomaly and background-only hypotheses.

A classical SPRT assumes known and constant model parameters. In reality, such parameter values are not always available, and thus we consider a *generalized likelihood ratio test* (GLRT) [37], defined as

$$G_N(\mathbf{x}) = \prod_{k=1}^N \frac{p(x_k, \hat{\Theta}_1|H_1)}{p(x_k, \hat{\Theta}_0|H_0)} \quad (4)$$

where we use the notation $p(x_k, \hat{\Theta}_i|H_i)$ to denote replacing the true values of the model parameters Θ_i of the conditional probability density $p(x_k|H_i)$ with their maximum likelihood (ML) estimates $\hat{\Theta}_i$. To form the *generalized SPRT*, the estimated parameters are substituted into the test form as previously described. In particular, we continue taking observations if $A < G_N(\mathbf{x}) < B$, and make a decision, choosing H_0 or H_1 if $G_N(\mathbf{x}) \leq A$ or $G_N(\mathbf{x}) \geq B$, respectively. When implementing the GLRT, the model parameters associated with either or both densities may be estimated. We adopt the notation $\hat{\theta}_i = \hat{\theta}|H_i$ to denote the estimate $\hat{\theta}$ of the parameter θ when H_i is true. Herein, for both the presence-of-anomaly and background-only hypotheses, the respective model parameters are estimated using the observations in the SPRTs for both our features.

In particular, the model parameters are updated using non-overlapping windows. We initially use fixed-size windows for both hypotheses; a 1-second sliding window ensures that enough data is being collected to derive good estimates of the background and attack parameters, denoted $M_{init} = N_{init} =$

1 second. The offset window employed to estimate the H_1 parameters uses more recent samples, and thus the change in the model parameters can be detected as evidenced in Section V. Whenever the SPRT crosses the lower threshold, confirming the absence of an attack, the ASN function (see Appendix B) is computed under hypothesis H_0 , and the update window size is reset to

$$M = \min \{ \mathbb{E}_0(N), M_{init} \}. \quad (5)$$

Similarly, when an attack is detected by the bPDM, the length of the update window for the H_1 parameters is reset to

$$N = \min \{ \mathbb{E}_1(N), N_{init} \}, \quad (6)$$

where the first argument of the min functions in (5) and (6) are the ASN functions under hypotheses H_0 and H_1 , respectively, and have been derived in Appendix B. We now derive the SPRTs for both the packet-rate and packet-size features, and then describe the bPDM algorithm.

A. The SPRT for the Packet-Rate

The null hypothesis H_0 , which represents only background traffic, is modeled using the generalized Poisson distribution (GPD), whose probability density function (pdf) is given by

$$p(x|H_0) = \theta(\theta + \lambda x)^{x-1} e^{-\theta - \lambda x} / x!, \quad (7)$$

where $x \in \{0, 1, \dots\}$ is the number of packet arrivals in a fixed time interval and $\{\theta, \lambda\}$ are the parameters of the GPD. We model an anomaly or attack stream as a constant-rate source with *deterministic, unknown* rate r . Our work focuses on detecting a set of commonly-occurring attacks, that is a class of attacks such as DoS attacks which use fixed-size attack packets [14]. Since DoS attacks are also characterized by the attacker flooding the network, this set of attacks corresponds to the constant-rate attack traffic assumption made above. However, as evidenced in Section V-E, the bPDM can also quickly and accurately detect smart attacks, which employ varying packet sizes. A random variable Y drawn from the anomalous distribution is specified as

$$Y = r + X, \quad (8)$$

where X is drawn from the GPD distribution that models the background only hypothesis. For the anomaly hypothesis, we assume that the constant-rate anomaly follows the pdf of the shifted GPD (sGPD)¹ given by

$$p(x|H_1) = \theta(\theta + \lambda(x - r))^{x-r-1} e^{-\theta - \lambda(x-r)} / (x - r)! \quad (9)$$

where $x \in \{r, r + 1, \dots\}$ is the number of packet arrivals in a fixed time interval and $\{\theta, \lambda, r\}$ are the parameters of the sGPD. Note that in the case where an anomaly is present, r is the minimum number of packet arrivals in a fixed time interval. For the packet-rate SPRT, under both the GPD and sGPD, x_i is thus the number of packet arrivals in the interval $\left[\frac{i}{p}, \frac{i+1}{p} \right)$, given the sampling rate p .

¹In our previous work [34], we modeled the presence of an anomaly using the simpler shifted Poisson distribution. The richer, generalized Poisson model is employed herein to circumvent the need for a dedicated training phase, and to allow all the model parameters to be estimated online.

The SPRT, in the case of the packet-rate feature, requires us to compare the generalized likelihood ratio

$$G_N(\mathbf{x}) = \prod_{k=1}^N \frac{p(x_k, \hat{\theta}_1, \hat{\lambda}_1, \hat{r} | H_1)}{p(x_k, \hat{\theta}_0, \hat{\lambda}_0 | H_0)} \quad (10)$$

to the threshold given in (3). Note that the densities specified in (10) are the GPD (7) and sGPD (9) with parameter estimates used in lieu of known parameter values. We now derive the estimator structures for the parameters of the GPD and sGPD for the background-only and presence-of-anomaly hypotheses, respectively.

The mean and variance of the GPD are given as [10]

$$\mu = \theta(1 - \lambda)^{-1} \quad \text{and} \quad \sigma^2 = \theta(1 - \lambda)^{-3}, \quad (11)$$

and are used to derive the moment estimators of the parameters θ and λ under the H_0 hypothesis, which are given as [10]

$$\hat{\theta}_0 = \sqrt{\frac{\bar{x}^3}{s^2}} \quad \text{and} \quad \hat{\lambda}_0 = 1 - \sqrt{\frac{\bar{x}}{s^2}}, \quad (12)$$

where \bar{x} and s^2 are the sample mean and sample variance, respectively, of an M -sample window. We note that the sample mean and sample variance are computed using their unbiased estimators² given by

$$\bar{x} = \frac{1}{M} \sum_{i=1}^M x_i \quad \text{and} \quad s^2 = \frac{1}{M-1} \sum_{i=1}^M (x_i - \bar{x})^2, \quad (13)$$

respectively. Although both moment and ML estimators are available (see [10] for the ML estimators) in the case of the null hypothesis, we use the former since they are more computationally efficient than the latter.

For the sGPD, the moment estimators of the three model parameters (θ_1, λ_1, r) require computing third- and fourth-order moments which we observed to required an order of magnitude greater number of samples to compute than the average time to detection. Thus, we present an alternative, computationally lightweight, estimation procedure for the model parameters under the H_1 hypothesis.

From the construction of our anomaly model in (8), we would expect that an unbiased estimate of r can be obtained by simply using the difference in the average traffic levels in the anomalous and background-only cases. Furthermore, since we are deriving moment estimators in a SPRT framework, we employ the estimator

$$\hat{r} = \max \left\{ \left\lfloor -\hat{\theta}_0 / (1 - \hat{\lambda}_0) + \bar{x} \right\rfloor, \min \{ x_1, \dots, x_M, x_{M+1}, \dots, x_{M+N} \} \right\} \quad (14)$$

where $\hat{\theta}_0$ and $\hat{\lambda}_0$ are as defined in (12). The estimate \hat{r} is computed using both the N -sample sliding window and the M -sample growing window. Since $Y - r$ is a generalized-Poisson-distributed random variable, the other two sGPD model parameters are estimated using the GPD estimator structures in (12) and are given as

$$\hat{\theta}_1 = \sqrt{\frac{(\bar{x} - \hat{r})^3}{s^2}} \quad \text{and} \quad \hat{\lambda}_1 = 1 - \sqrt{\frac{\bar{x} - \hat{r}}{s^2}} \quad (15)$$

²An estimator is defined as *unbiased* if the estimator's expected value is equal to the true value of the parameter being estimated, i.e. $\mathbb{E}\{\hat{\theta}\} = \theta$ [23].

where \hat{r} is the estimate given in (14), and \bar{x} and s^2 are computed via (13) using the N -sample sliding window. Note that the $\min\{\cdot\}$ function in (14) is a constraint due to the fact that the support of the sGPD is over $\{r, r+1, r+2, \dots\}$. We further note that the derived estimators $(\hat{\theta}_1, \hat{\lambda}_1, \hat{r})$ provide accurate estimates of the sGPD parameters, and are *consistent*³ (refer to Appendix D of [35]).

Employing the GPD/sGPD hypothesis test lets us detect a change in the mean of the traffic, but an increase in the mean does not always occur due to a malicious anomaly or attack. Flash crowds, which might occur due to the Digg or SlashDot effect, are not malicious traffic [20], but would be tagged as anomalous.

B. Incorporating the Packet-Size SPRT

The packet-size distribution of normal Internet traffic has been characterized in [28] as mostly bimodal at 40 bytes and 1500 bytes (with 40% and 20% of packets, respectively). An examination of our background trace data, which include Ethernet and VLAN headers, validates this model but with differing means. The background traffic in our traces can also be characterized as mostly bimodal, with means at 68 bytes and 1518 bytes, which represent approximately 40% and 20% of the total packets, respectively. We note, however, that no specific distribution is ascertained for the remaining 40% of the packets.

We expect packet-size distribution information to be effective in attack detection, since a broad class of attacks use a single packet-size; *e.g.* DNS reflector attacks use the maximum packet-size and TCP SYN attacks use the minimum packet size. Thus, the influx of attack packets, in the case of attacks that employ a single attack packet size, will alter the relative number of a specific packet-size with respect to the packet-size distribution of normal traffic. As such, the sample entropy of the packet-size distribution can be used to distinguish between the background-only and presence-of-anomaly hypotheses.

In the bPDM framework, recall that x_i represents the number of packet arrivals in the interval $\left[\frac{i}{p}, \frac{i+1}{p}\right)$. Let \mathbb{S}_i denote the set of distinct packet sizes that arrive in this interval, and q_j denote the proportion of packets of size j to the total number of packets in the same interval. Thus, the sample entropy y_i is computed as

$$y_i = - \sum_{j \in \mathbb{S}_i} q_j \log q_j . \quad (16)$$

The sample entropy is modeled using the Gaussian distribution given by

$$p(y|H_i) = \frac{1}{\sqrt{2\pi\sigma_i}} \exp \left[-\frac{1}{2\sigma_i^2} (y - \mu_i)^2 \right] \quad (17)$$

for both the background ($i = 0$) and attack ($i = 1$) hypotheses. Thus, the log-likelihood ratio (LLR), given N observations, is

³A sequence of estimators for a parameter θ is said to be consistent (or asymptotically consistent) if the sequence converges in probability to θ . In our case, the estimator is a function of the sample sizes M and N . Thus, as M and N tend to infinity, the estimator converges in probability to the true value of the parameter, and the mean-squared error tends to zero [12].

specified as

$$\log L(\mathbf{y}) = a_2 \sum_{i=1}^N y_i^2 + a_1 \sum_{i=1}^N y_i + a_0, \quad (18)$$

where $a_2 = \frac{1}{2\sigma_0^2} - \frac{1}{2\sigma_1^2}$, $a_1 = \frac{\mu_1}{\sigma_1^2} - \frac{\mu_0}{\sigma_0^2}$, and $a_0 = N \left[\frac{\mu_0^2}{2\sigma_0^2} - \frac{\mu_1^2}{2\sigma_1^2} + \log \left(\frac{\sigma_0}{\sigma_1} \right) \right]$. As in the case of the GPD/sGPD hypothesis test, the model parameters in the case of the sample entropy are estimated in real time using the sliding and growing update windows. Since the sample entropy is modeled using the Gaussian distribution, the parameter estimators for μ and σ^2 for each of the hypotheses are the sample mean \bar{x} and sample variance s^2 , given in (13), using the respective update windows. The resulting SPRT requires that we continue to take more observations if

$$\log(A) < \log G(\mathbf{y}) < \log(B), \quad (19)$$

where $G(\mathbf{y})$ is the generalized likelihood ratio associated with the packet-size SPRT. $\log G(\mathbf{y})$ is of the form in (18), but the constants a_2, a_1 and a_0 are defined in terms of the parameter estimates $\{\hat{\mu}_0, \hat{\sigma}_0^2\}$ and $\{\hat{\mu}_1, \hat{\sigma}_1^2\}$ instead of the true parameter values.

Given two features, ideally we would compute a joint density to determine a single bivariate SPRT. However, given the mixed nature of the two features (discrete packet arrivals and continuous entropies) computing this joint density appears to be intractable. Instead, we now describe our bPDM algorithm, which effectively combines the two SPRTs to yield an anomaly detection mechanism that has a low probability of false positives.

C. The bPDM Algorithm

The bPDM combines the SPRTs of the packet-rate and packet-size features. Before we present the bPDM implementation details, we first consider a pedagogical example that shows that a fixed-size DoS attack can be successfully detected by the bPDM, which combines both the packet-rate and packet-size features. Given the bimodal characterization of packet sizes [28], assume that normal background-only traffic has a packet-size distribution given by:

$$\left[\begin{array}{l} 40\% \text{ 68-byte packets} \\ 20\% \text{ 1518-byte packets} \\ 40\% \text{ other packets} \end{array} \right].$$

We further assume that the DoS attack, which uses 1518-byte packets, increases the percentage of 1518-byte packets from 20% to 40%. Assuming that the background-only traffic consists of 100 packets, we can calculate that the attack consists of 33 packets, which result in the increased proportion of 1518-byte packets. We can now compute the post-attack distribution of packet sizes as:

$$\left[\begin{array}{l} 30\% \text{ 68-byte packets} \\ 40\% \text{ 1518-byte packets} \\ 30\% \text{ other packets} \end{array} \right].$$

Thus, we see that volume-based, fixed-size DoS attacks can be successfully detected by the bPDM since the attack alters both the packet-rate and packet-size distributions.

For the implementation of the bPDM, we first recall that the bPDM must be initially deployed in the absence of an anomaly. Once the initial parameter estimates have been computed, subsequent observations are used to update the parameter estimates for both hypotheses and compute the likelihood ratios. For each of the SPRTs, the likelihood ratio is updated given each new observation as described in (2). The continuous updating of the likelihood ratio and the H_0 and H_1 parameters, estimated using a fixed number of samples, obviates the necessity of *a priori* knowledge of the background or baseline parameters.

During the operation of the bPDM, if only one of the SPRTs (packet-rate or packet-size) crosses the upper threshold B , then we declare an initial warning and continue computing the likelihood ratio after resetting the corresponding SPRT. For example, an increase in the packet rate without a significant change in the sample entropy of the packet-size distribution may be due to a normal non-malicious increase in traffic. Thus, an attack is declared only if an initial warning is followed by the other SPRT crossing the upper threshold; *i.e.*, we declare an attack only if *both* the packet-rate and packet-size SPRTs “coincidentally” cross the upper threshold. Requiring the SPRTs to cross the upper threshold at the same sample is too restrictive, being the equivalent of millisecond accuracy; thus, we define the “hold time,” $\tau_H = 0.1$ second, and require that the SPRTs cross the upper threshold within $\tau_H p$ samples of each other. Consequently, a false positive is said to have occurred when both SPRTs coincidentally cross the upper threshold *and* there is no anomaly present in the traffic. The salient features of bPDM operation are highlighted in Figure 2, and pseudocode for the algorithm can be found in our technical report [35]. In contrast to the bPDM operation described, the packet rate could also be used individually to detect anomalies. However, this would result in a significant number of false positives being declared since most legitimate increases in traffic would be flagged as attacks.

D. An Example and Generalization

In order to highlight the facets of the bPDM, we consider the detection of a simulated Iperf attack with bitrate SNR⁴ 0.056 using our detection mechanism. Figure 3 shows the SPRT outputs that result from detecting this Iperf attack. We declare the presence of an attack 695 msec after the start of the attack, although the packet-size SPRT crosses the upper threshold before this point in time. The delay is due to the fact that the bPDM requires that both SPRTs coincidentally cross the upper threshold. Although it is the case that the time to detection could have been reduced had we used only the packet-rate SPRT in the case of Figure 3, it would have yielded a false positive in the case of Figure 4. Therein, we find that the crossing of the upper threshold before the start of the attack is flagged as a warning, but an attack is not declared. Thus, the bPDM reduces false positives by leveraging both the packet-rate and packet-size features of aggregate traffic.

⁴The bitrate SNR, defined in (20), is an SNR measure – a metric that is used to compare the relative strengths of two attacks.

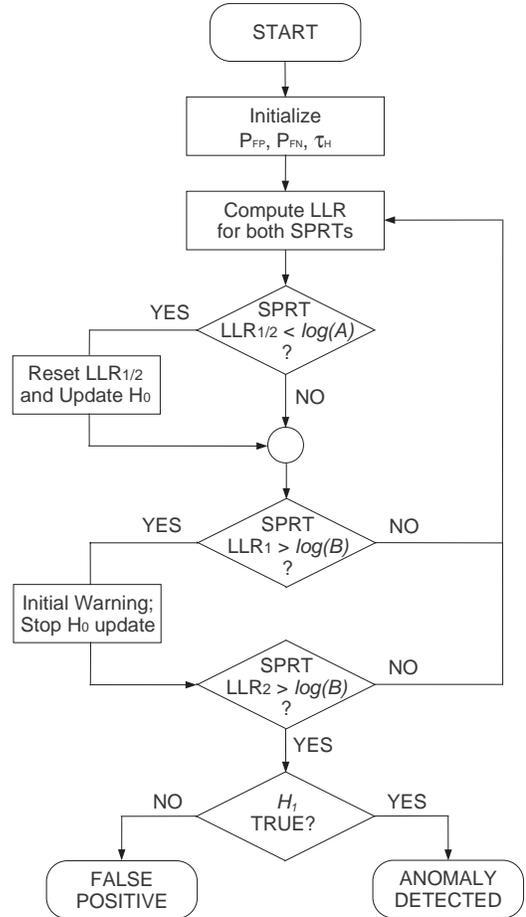


Fig. 2. Flowchart highlighting the salient features of the operation of the bivariate Parametric Detection Mechanism (bPDM).

An advantage of anomaly detection based on aggregate packet rates and sizes, instead of contents, is that it is robust to encryption. A DoS attack with encrypted traffic will show the same rate of change in packet sizes as during a DoS attack. These results assume that encryption is packet-length preserving (as is typical for nearly all network encryption schemes). While our extensions that consider packet size would be ineffective for encryption that performed traffic obfuscation and bundling, our rate-based methods apply even there.

V. PERFORMANCE EVALUATION AND ANALYSIS

In the following sections, we employ three sets of synthetic traces, six real and proxy-real network attacks, and 67 emulated Iperf attacks in varying traffic mixes to investigate the effects of background and attack traffic levels on the time to detection. We show that the performance of the bPDM is comparable to or better than selected alternate detection schemes (Section V-C), and that time to detection is influenced by bitrate SNR. We define bitrate SNR in Section V-A, and show how it is affected by attack rate and hopcounts in Sections V-B and V-H, respectively. We also compare it to the previously-used packet SNR metric (Section V-F). Finally, we validate our synthetic attacks (Section V-G), and demonstrate

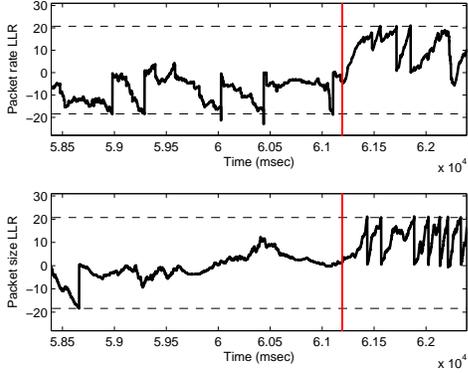


Fig. 3. SPRTs for the packet-rate and packet-size features for the Iperf attack with bitrate SNR 0.056: an attack is declared when *both* SPRTs cross the upper threshold B , 695 msec after start of attack.

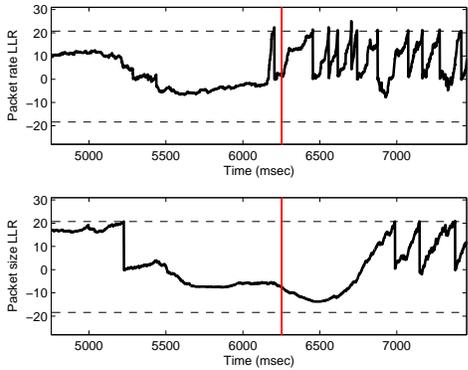


Fig. 4. SPRTs for the packet-rate and packet-size features for a synthetic TCP SYN attack with bitrate SNR 0.005. Non-coincidental crossings are simply flagged as warnings.

that the bPDM works with minimal training (Section V-D), is robust to countermeasures (Section V-E), and has a controllable probability of false positives (Section V-I).

A. Evaluation of the bPDM

The basic principles of detection theory teach that the time to detection for a signal in noise is related to the signal-to-noise ratio (SNR) [37]. However, for anomaly detection, there is no clear notion of what an appropriate SNR measure would be. We present the *bitrate SNR* metric, which is defined as

$$\text{bitrate SNR} = \frac{\text{Anomalous traffic level}}{\text{Background traffic level}} = \frac{\sum_{S \in \mathcal{S}_A} M_S S}{\sum_{S \in \mathcal{S}_B} M_S S}, \quad (20)$$

where \mathcal{S}_A is the set of attack packet-sizes, \mathcal{S}_B is the set of background packet-sizes, and M_S is the number of packets of size S bits.

In this section, we evaluate the bPDM using a set of synthetic traces and emulated Iperf attacks, and find that as the bitrate SNR increases, the time to detection decreases. This trend is also shown to be true for the underlying theoretical model of the bPDM.

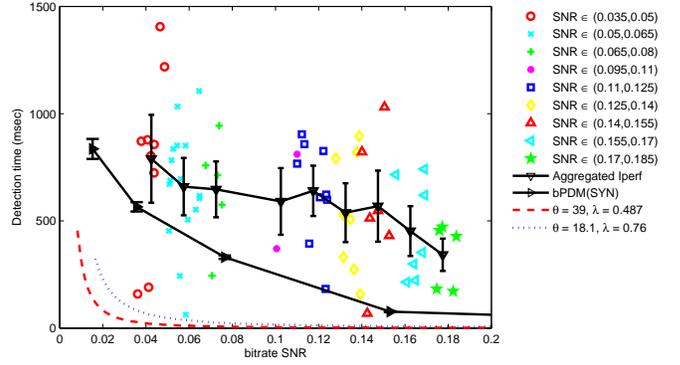


Fig. 5. Comparing detection time for the emulated Iperf attacks and the synthetic TCP SYN attacks, with the Iperf attacks grouped by similar bitrate SNR values. Theoretical detection times also plotted for comparison.

1) *Evaluation using simulated synthetic traces:* The bPDM is first evaluated using a set of synthetic attacks [5] that allow us to control the attack rate and methodically evaluate the bPDM. The attack traces use 196 megabits per second (Mbps) background traffic taken from our network. After 6–8 seconds of background traffic, we add in constant-rate attacks at various rates using Stream Merger [22]. Focusing on low-rate attacks, our traces employ attacks that range from 1 Mbps to 120 Mbps, in addition to the 196 Mbps background traffic. The artificial attacks model TCP SYN attacks that use a fixed-size attack packet of 68 bytes [8].

Figure 5 plots the bPDM times to detection for the set of synthetic TCP SYN attacks as a function of the bitrate SNR, in addition to the detection times for emulated attacks and the theoretical model discussed in following sections.

The bPDM was run on 8 distinct synthetic traces of a specific bitrate SNR, and the mean values of the detection times are plotted in Figure 5 along with error bars that represent the standard deviation associated with the mean detection time. We see that as the bitrate SNR increases, the bPDM time to detection for the synthetic attacks decreases.

2) *Evaluation using emulated Iperf traces:* We next consider a more realistic scenario wherein controlled attacks in *varying* traffic mixes are detected by our algorithm. Specifically, we employ 80-second Iperf attacks that use 345-byte fixed-size packets sent from Colorado State University (CSU) to the University of Southern California (USC); their generation is detailed in Appendix A. As before, as the bitrate SNR increases, the time to detection for these emulated Iperf attacks decreases.

The detection times for the 67 individual Iperf attacks are plotted as open symbols in Figure 5. The individual attacks are grouped by bitrate SNR to allow us to investigate the relationship between detection time and bitrate SNR. The data is partitioned in 0.015 bin increments, so that ten bins span the bitrate SNR range from 0.035 to 0.185. Each bin is plotted using a different symbol in Figure 5, *i.e.* data points that have bitrate SNR values between 0.035 and 0.05 are represented using red circles, data with bitrate SNR values between 0.05 and 0.065 by cyan x's, and so on. The aggregated Iperf line in Figure 5 plots the mean of each bin, and the error bars give

the standard deviation for each bin. We find that the time to detection decreases as the bitrate SNR increases for these Iperf attacks, as it did for the synthetic attacks. The large error bars of the aggregated Iperf plot prevent further statistical analysis.

As shown in Figure 5, there are no detection times that correspond to a bitrate SNR of less than 0.02 for both the emulated Iperf attacks, as well as the synthetic TCP SYN attacks considered previously. This constitutes a lower limit on the performance of the bPDM, in that if the attack rate is lower than 0.02, then the estimate of the r parameter of the shifted Generalized Poisson distribution in (14) is zero, and thus the attack is undetectable by the bPDM.

3) Comparing simulated and emulated traces to theory:

We have found that the bPDM time to detection decreases as the bitrate SNR increases in the case of both the simulated and emulated attacks. In this section, we show that the time to detection for the underlying theoretical model follows the same general trend. We recall that the sequential probability ratio test (SPRT), described in Section III, is employed by the bPDM for both the packet-rate and packet-size features. For the packet-rate SPRT, which is based on the generalized Poisson distribution (GPD) model as in (7) and (9), the theoretical time to detection is the average sample number (ASN) function under hypothesis H_1 , and is derived in Appendix B. The ASN under H_1 is a function of the shifted GPD (sGPD) model parameters $\{\theta, \lambda, r\}$.

The same set of sGPD parameters is used to derive the bitrate SNR, as defined in (20). The mean of the GPD is $\theta/(1-\lambda)$ (see (11)), which corresponds to the number of packets in the background traffic. Similarly, the attack parameter r corresponds to the number of attack packets. Furthermore, we assume that the attack uses constant 544-bit packets, and adopt a simplified model for the background traffic wherein 66.6% of packets are 480-bit, and 33.3% packets are 12000-bit. Thus, the bitrate SNR is computed as

$$\text{bitrate SNR} \Big|_{\text{bPDM}} = \frac{r \cdot 544}{(2/3 \cdot 480 + 1/3 \cdot 12000)\theta/(1-\lambda)}, \quad (21)$$

where for a fixed θ and λ , a greater r corresponds to a higher bitrate SNR. The theoretical detection times for $\{\theta = 39, \lambda = 0.487\}$ and $\{\theta = 18.1, \lambda = 0.76\}$, which correspond to the parameter values for the synthetic TCP SYN attacks and a 30 Mbps Iperf attack, respectively, are plotted in Figure 5 as a dashed red line and a dotted blue line, respectively.

We see that the theoretical time to detection trends as in the experimental cases: the time to detection decreases as the bitrate SNR increases. In the theoretical case, we find that the detection time is an exponential function of the bitrate SNR; lower-rate attacks take significantly longer to detect than high-rate ones. The theoretical detection times are much lower than the empirical times since there is *no* notion of cross traffic, or interaction between the packets from the background and attack streams, as experienced in a real router (in the case of the Iperf attacks) or in the Stream-Merger application (in the case of the synthetic TCP SYN attacks). A similar trend is seen in the case of the experimental data, but a rigorous fit cannot be performed due to the small set of averaged data points. Thus, we find that attacks with higher bitrate SNR values are

detected more quickly for the simulated and emulated attacks, which is consistent with what is predicted by the underlying theoretical model.

B. Effect of Attack rate (Mbps) on Time to Detection

In the previous section, we saw that the time to detection decreases as the bitrate SNR increases. The bitrate SNR as defined in (20) consists of two components: the attack rate (in Mbps) and the background traffic (in Mbps). We now investigate the effect of each of the individual components on the time to detection, and find that for a constant level of background traffic, the time to detection decreases as the attack rate increases. The effect of varying background traffic for a constant attack rate is considered in the next section.

As in Section V-A2, we again aggregate the emulated Iperf data, this time to better examine the effect of the attack rate. Specifically, we group the detection times of emulated attacks by level of background traffic: data points with background traffic of less than 350 Mbps constitute the first group (low-level), and data points with background traffic greater than 350 Mbps are the second group (high-level)⁵. The detection times of the Iperf attacks, grouped by background traffic level, are plotted as a function of the attack rate (in Mbps) in Figure 6. The attack rates for the data points in Figure 6 are all at 20, 25, 30 or 40 Mbps, but are plotted with random shifts ($\in (-1, 1)$ Mbps) to improve the visibility of the data points.

To measure the association of the time to detection with the attack rate, we compute the Pearson product-moment correlation coefficient r [32] of the time to detection and background traffic level, as well as the time to detection and attack strength. The correlation coefficient is independent of the scale of measurement, and its value ranges from -1.00 to +1.00⁶, wherein an r value of 0.00 represents no correlation between the two variables, while a value of -1.00 or +1.00 indicates perfect predictability.

For this grouping of the data points, the correlation coefficients between detection time and attack rate (Mbps) for the high- and low-level background traffic are -0.3050 and -0.0781 , respectively. The correlation coefficients and their associated p -values, along with the sample size for each group, are listed in Table I. The p -value is a measure of statistical significance, *i.e.* the probability that the result occurred due to chance rather than an underlying cause. A p -value of less than 0.10 indicates that there is statistical evidence for the model being considered, or hypothesis being proposed, at the 10% significance level. We see that the detection time and attack rate are weakly negatively correlated with statistical significance for $R_{bg} < 350$, suggesting that for a specific background level of traffic, the time to detection decreases as the attack rate increases.

⁵Finer groupings of background traffic (100-200 Mbps, 200-300 Mbps, etc.) were also considered, but the results were inconclusive due to insufficient data points in each bin.

⁶Given the variables Y and X , we define the standardized variables $Z_Y = (Y - \bar{Y})/S_Y$ and $Z_X = (X - \bar{X})/S_X$, where (\bar{Y}, \bar{X}) and (S_Y, S_X) represent the sample means and standard deviations of the variables Y and X , respectively. The Pearson r is then computed as $r = \sum Z_X Z_Y / (N - 1)$ [32].

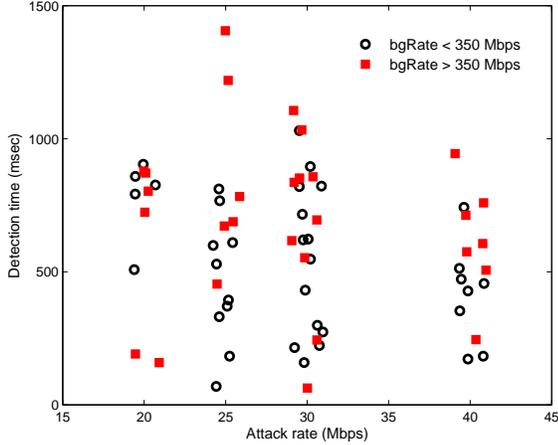


Fig. 6. Detection time for the Iperf attacks as a function of attack rate (in Mbps), grouped by high and low levels of background traffic.

TABLE I
CORRELATION COEFFICIENTS AS A FUNCTION OF BACKGROUND RATE,
 R_{bg} .

Bin (Mbps)	$R_{bg} < 350$	$R_{bg} > 350$
Sample size	37	30
r	-0.3050	-0.0781
p -value	0.0664	0.6815

Note that this weak negative correlation between time to detection and attack rate also holds in the case of the set of synthetic TCP SYN attacks discussed in Section V-A1 above, where the time to detection decreased as the bitrate SNR increased. This result is intuitive because in both cases, the effect of attack rate is examined for a given level of background traffic.

Interestingly, for a higher level of background traffic ($R_{bg} > 350$), the correlation becomes very weak ($r = -0.0781 \sim 0$) and loses statistical significance ($p = 0.68 > 0.10$). In order to support the claim that the decrease in correlation is reflective of a legitimate trend, and not merely an artifact, we now consider the variances of the detection times, grouped by background traffic level, as a function of the attack rate. Table II shows that for the higher-level background traffic, the variance in detection time is greater than in the low-level background traffic case for all the attack rates. Recall that the theoretical cases considered in Section V-A suggest that the time to detection is an exponential function of the bitrate SNR. This trend can also be seen, but not rigorously verified, for the experimental synthetic and Iperf data. However, the statistical analyses presented herein show that as the level of background traffic increases, the attack rate is less predictive of the time to detection of the bPDM.

C. Comparing bPDM to Prior Methods

We compare the bPDM to a selected set of detection schemes as described in Section II, and find that our algorithm performs comparably to or better than the other detection mechanisms we consider, while mitigating key drawbacks of

TABLE II
VARIANCE OF DETECTION TIMES AS A FUNCTION OF BACKGROUND RATE,
 R_{bg} , GROUPED BY ATTACK RATE R_{att} .

R_{att} (Mbps)	40	30	25	20
$R_{bg} < 350$ Mbps	$3.4e4$	$8.15e4$	$5.8e4$	$2.44e4$
$R_{bg} > 350$ Mbps	$4.8e4$	$10.0e4$	$13.2e4$	$11.4e4$

the latter. Recall that the bPDM only requires 2-3 seconds of background-only traffic for training, updates its model parameters in real-time, and requires no human intervention when it is initially deployed. First, we focus on the Modeled Attack Detector (MAD) [34], a time-domain sequential scheme which adopts a simpler Poisson model and only uses the packet-rate feature to detect attacks. The MAD requires at least 10–12 seconds of background-only data to initialize the estimate of its background parameter λ (the rate), which does not update during the algorithm’s operation. Though in the MAD the attack parameter r (the rate change due to the attack) is updated in real time, the fact that the background parameter remains static necessitates a longer training phase as compared to the bPDM, which requires 2–3 seconds of training data. Furthermore, significant evolutions of normal network traffic are often flagged by the MAD as attacks since the background parameter is not automatically updated. In contrast, the bPDM updates its model parameters in an online fashion, and employs the packet-size feature to minimize false alarms.

The second scheme we consider is the Periodic Attack Detector [34], a spectral-domain scheme that exploits the near-periodic nature of attacks. The PAD is the sequential version of the spectral-domain scheme developed by He et al. [17], *i.e.* the underlying models and development in [17] were adapted into a sequential framework as described in Section III. Like the MAD, the PAD uses a longer training interval compared to the bPDM, and additionally requires that the test data be statistically similar to the training data.

Unlike the bPDM, which develops statistics based only on aggregate traffic features, the entropy-based scheme by Feinstein et al. [15] computes the entropy of *flow-separated parameters* and compares the decision statistics against a threshold in a sequential framework. We simulate this scheme by computing the entropy of the destination IP address using non-overlapping batches of 5000 packets, and note that the time to detection for this method, listed in Table III, does not include the time required to extract the flow-separated parameters. We reiterate that the bPDM detects attacks, and that flow-separation would then be required to filter out the attack. In effect, the proposed approach serves as a very *light-weight*, early attack detection mechanism. If flow-separation is required, the same method used in Feinstein et al. can be used for the bPDM and thus will take the same amount of time. The key difference is that Feinstein’s detection method requires *a priori* flow-separation, whereas our method would only invoke flow-separation after attack detection.

We compare the performance of these three schemes to the bPDM using the bitrate SNR metric. First, we compare the four methods’ performance when tested on a reflector attack [4] with a bitrate SNR of 0.0678, which sends echo reply

TABLE III
NUMERICAL RESULTS FOR COMPARISONS OF THE bPDM TO OTHER METHODS.

Scheme	# FP	TD	Drawback
bPDM	0	336	Limited training required
MAD [34]	2	280	Longer training phase required
IP Entropy [15]	0	400	Flow-separation required
PAD [17]	1	340	Higher complexity due to FFT and longer training phase required

packets targeting a victim within Los Nettos and lasts for 204 seconds. The results are tabulated in Table III, where the second and third columns are the number of false positives (# FP) and the time to detection (TD, in msec), respectively. We note that the detection time for the method by Feinstein et al. [15] may be shorter or longer if different simulation parameters are employed. Comparison of the four methods shows that the time to detection for the bPDM is comparable to or shorter than those of the other three.

Next, we employ the set of synthetic TCP SYN attacks to compare the MAD and PAD detection schemes to the bPDM. Figure 7 shows the detection time as a function of the bitrate SNR for the bPDM, MAD, and PAD schemes. The IP entropy scheme [15] is not included in the comparison because the synthetic attacks were generated without source and destination IP addresses and port numbers. The label “bPDM(SYN)” denotes the performance of the bPDM on the set of synthetic TCP SYN attacks, and similarly for MAD and PAD. Each of the three algorithms was run on 8 distinct synthetic traces of a specific bitrate SNR, and the resulting mean values are plotted in Figure 7 along with error bars representing the standard deviation associated with the mean detection time for each bitrate SNR. Notice that, as expected, the time to detection decreases as the bitrate SNR increases. We find that the bPDM generally outperforms both the MAD and the PAD, although for lower bitrate SNR values, its detection times are comparable to the MAD’s. The spectral-based PAD consistently has the highest, though comparable, detection times. We note that we achieve these comparable or better detection times, in the case of fixed-size constant-rate attacks, without the drawbacks of the other methods, as described in Sections I and II (see Table III). For the case of the smart attacker (see Section V-E), the detection times for the bPDM can be considerably longer.

D. Validating the bPDM’s need for only minimal training

The previous section showed that the bPDM outperforms the MAD, PAD, and entropy-based Feinstein schemes. In this section, we further explain two important advantages of the bPDM: it requires limited training, and its model parameters automatically update in an on-line fashion, as compared to other existing schemes.

The MAD (see Section V-C for details) requires a 10–12 second or more training period and flags both marked changes in the level of background traffic and actual attacks as attacks, since the increase in traffic volume is captured by updating only the attack parameter. In contrast, the bPDM uses the generalized Poisson distribution with two varying parameters

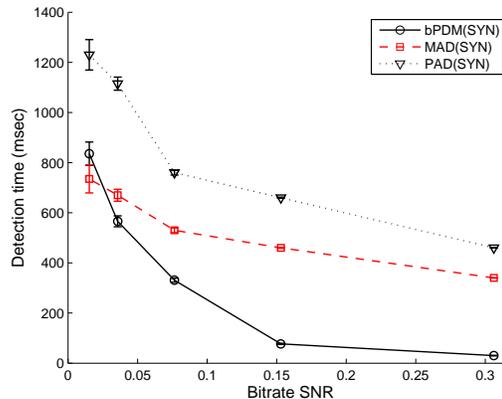


Fig. 7. Comparing the time to detection (in msec) for the bPDM, MAD and PAD detection algorithms using the set of simulated synthetic TCP SYN attacks that employ 68-byte packets.

(θ , λ) to account for changes in the level of background traffic. The attack parameter r , as in the case of the MAD, updates as in (14), and is compared to the background to identify the presence of an anomaly. Importantly, in the bPDM, the packet-size SPRT is incorporated in addition to the packet-rate SPRT to ensure that an increase in traffic volume without a corresponding change in the packet-size distribution is *not* flagged as an attack (see Section IV-C and Figure 2 for details). We note that even if the packet-size SPRT had likewise been incorporated into the MAD to reduce false alarms, the static nature of the background parameter would still necessitate a minimum 10–12 second training period for the MAD, while the bPDM only requires 2–3 seconds. Furthermore, our experimental results suggest that the bPDM requires only 2–3 seconds of background traffic *irrespective* of the attack strength.

The second detection scheme to be considered, the PAD, also requires a longer period of training data than the bPDM, which it uses to characterize the spectral-domain features of normal background-only traffic. In the testing phase, the presence of frequency-domain components that were not present in the background-only traffic spectrum is used to detect attacks. We note that the PAD is sensitive to significant changes in the background traffic, and thus the data used to train the algorithm must be statistically similar to those used to test it. This is not the case for the bPDM, which initializes using a limited amount of training data, and then automatically updates its parameters as the network traffic evolves.

The MNA-CUSUM is a non-parametric sequential algorithm developed by Tartakovsky et al. [33], which requires a non-trivial amount of overhead when initially deployed: it filters incoming packets by size and uses individual channels and decision statistics (analogous to the log-likelihood ratio in the SPRT) to detect an attack rapidly. Because the decision statistics are based on score functions that update periodically using a parameter update method similar to that of the bPDM for each of the channels, the initial deployment of the MNA-CUSUM involves hand-tuning of the thresholds of each channel to meet the false alarm requirements. An alternative

to the hand-tuning of thresholds could be an explicit search over the parameter space, which has not been implemented in [33] but would presumably be computationally intensive due to the multiple channels employed. Recall that the bPDM is initially deployed with no hand-tuning, since the initial parameter estimates are automatically computed using (12), given a limited amount of background-only training data.

Thus, our algorithm requires only up to 2–3 seconds of training data, as compared to the 10–12 seconds needed by the MAD and PAD, since the H_0 and H_1 update window sizes are 1 second long as described in Section IV. And unlike MNA-CUSUM, our algorithm requires only a few parameters and performs automatic training. Thus, we find that the bPDM’s use of limited training data and automatic updating of its model parameters in real time results in its being free from the drawbacks of other, existing detection methods.

E. Robustness of the bPDM to a smart attacker

As described in previous sections, the bPDM uses the packet-size both as a feature for detection and to reduce false positives. We now consider the *smart adversary* scenario, wherein the attacker constructs an attack whose distribution of packet sizes attempts to match that of the background traffic. For this purpose, we create a set of smart adversary synthetic attacks wherein the attack stream uses a constant bitrate, but with a distribution of packet sizes that is drawn from the bimodal distribution described in [28]. We recall that the packet-size distribution of nominal Internet traffic has been characterized in [28] as mostly bimodal, a result validated by an examination of our background trace data. For any given data rate corresponding to a bitrate SNR, the smart attack is generated by combining 40% 68-byte packets, 20% 1518-byte packets, and a uniform distribution of the remaining 40% of packets in the interval (68,1518) bytes since the smart attacker is not privy to the exact (and evolving) distribution of packet sizes in background traffic at the detector’s location. Though attackers do not use this approach today, as in general they cannot perfectly guess the packet-size distribution on the monitored link, we present these smart attacks to consider one possible set of countermeasures against our detection mechanism.

It must be noted that, although the smart adversary actively manipulates only the packet-size distribution, the smart attacks affect both the packet-rate and the packet-size aspects of the bPDM:

- 1) The packet sizes used by a smart adversary are drawn from a bimodal distribution that resembles normal Internet traffic. This results in the entropy of the packet-size distribution in the case of an attack being similar to that in the background-only case, reducing the effectiveness of the packet-size SPRT.
- 2) Recall that the synthetic TCP SYN attacks employ 68-byte packets. In drawing from a bimodal distribution, the smart adversary uses a range of packets, including several that are larger than 68 bytes. This variety of packet sizes implies that for a fixed attack rate, say 60 Mbps, the smart attack has a smaller number of

packets per second, than a TCP SYN attack. This should challenge the packet-rate SPRT employed by the bPDM. Despite these challenges, we will see that the bPDM can still detect attacks from a smart adversary.

Figure 10(a) shows the bPDM detection times for the smart, denoted “bPDM(smart),” and TCP SYN, denoted “bPDM(SYN),” simulated attacks as a function of the bitrate SNR. We recall that the (SYN) label refers to the set of synthetic TCP SYN traces wherein fixed 68-byte packets are employed; correspondingly, the (smart) label denotes packets drawn from the bimodal packet-size distribution. In particular, the bPDM algorithm was run on 8 synthetic TCP SYN and smart traces each, all of a specific bitrate SNR. The mean detection times are plotted in Figures 10(a), wherein the error bars represent the standard deviation of the detection times. We find that the bPDM successfully detects the synthetic smart attacks, albeit with longer detection times than for the TCP SYN attacks.

Since the smart attacker employs the bimodal distribution of traffic, the sample entropy due to the packet sizes that correspond to the two means of the bimodal distribution does not change appreciably after the attack. On the other hand, the 40% of packets that have sizes that are uniformly distributed do not precisely model the true distribution of packet sizes of background traffic, and thus there is a measurable difference in sample entropy that enables the bPDM to detect the smart attacks. We now consider a smart attacker that generates a “smarter” attack that more closely matches the two most common packet sizes, and further incorporates the third (denoted *smart2*), and then the fourth (denoted *smart3*), most common (modal) packet sizes when designing the attack packet-size distribution. In this scenario, the original smart attack matches 58% of traffic, and the *smart2* and *smart3* attacks consist of 71% and 78% matched traffic, respectively. Figure 8 shows the packet-rate SPRT (subplot 1) and the packet-size SPRT for increasingly smarter attacks (subplots 2–4). Note that the packet-rate SPRT rapidly crosses the upper threshold for these attacks with bitrate SNRs of 0.3, whereas the packet-size SPRT takes longer since the attack packet size distribution more closely resembles that of background traffic. In the case of the “smarter” attack, wherein 58% of the traffic is matched by the attacker (corresponding to an empirical symmetric Kullback-Leibler (KL) divergence [11] of 14.21 with respect to the background), the bPDM time to detection is 770 msec (averaged over eight simulations). As the percentage of matched traffic increases, so does the time to detection. For the *smart2* attack, wherein 71% of the traffic is matched (empirical symmetric KL divergence of 9.42), the time to detection for the bPDM is 1500 msec (averaged over eight simulations). When 78% of the traffic is matched (empirical symmetric KL divergence of 7.33), as in the case of the *smart3* attack, the bPDM cannot detect the smart attacker. Even for the *smart3* attack, the packet-rate SPRT rapidly crosses the upper threshold at the onset of the attack (as in Figure 8, subplot 1), but an attack is never declared since the packet-size SPRT never crosses the upper threshold. Thus, although the packet-size SPRT significantly reduces false positives (especially in the case of fixed-size,

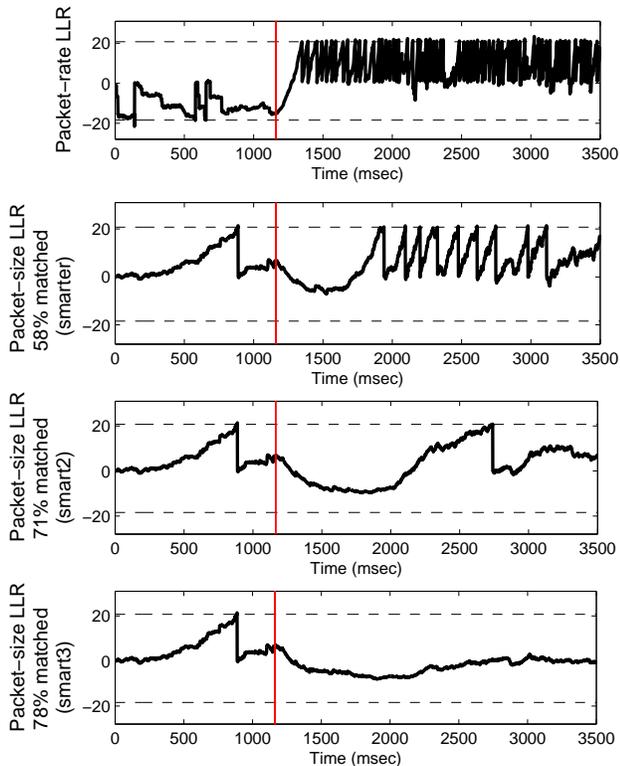


Fig. 8. Comparing the packet-size LLRs for increasingly “smarter” attacks; subplots 2–4 employ smart attacks that comprise 58%, 71% and 78% matched traffic, respectively.

constant-rate attacks), in the case of “smarter” attacks, it is responsible for false negatives. The consideration of the set of “smarter” attacks further characterizes the performance limitations of the bPDM. As seen in Section V-A1, the bPDM cannot detect attacks with bitrate SNRs lower than 0.02, and we now conclude that attacks with packet-size distributions that comprise roughly 80% or more matched traffic are also missed by the bPDM.

The bPDM’s inability to detect attacks with packet-size distributions that closely match that of the background traffic is also relevant in the case of open-loop versus closed-loop TCP flows. Since open-loop TCP flows exhibit greater traffic variability than closed-loop TCP flows [26], we would expect the packet-size distribution of the latter to resemble that of background traffic more closely than that of the former. In general, this would result in increased detection times for the bPDM for closed-loop TCP traffic. However, quantifying the difference in the packet size distributions for open-loop and closed-loop TCP traffic, and its effect on bPDM performance, is beyond the scope of this paper.

F. Bitrate SNR versus packet SNR

An alternative metric, the packet SNR, is used by He et al. [17] to evaluate their methods. In this section, we compare the packet SNR to the bitrate SNR, and we find that the latter is a more effective measure of anomaly strength for this

application. The packet SNR is defined as [17]

$$\text{packet SNR} = \frac{\# \text{ of attack packets}}{\# \text{ of background packets}} = \frac{\sum_{S \in \mathcal{S}_A} M_S}{\sum_{S \in \mathcal{S}_B} M_S}, \quad (22)$$

where \mathcal{S}_A , \mathcal{S}_B and M_S are as defined for (20). This metric is thus defined in terms of the packet rates of both the attack and the background traffic, rather than the bitrate (in Mbps) as in the case of the bitrate SNR.

Both the packet and bitrate SNRs are equivalent metrics for the TCP SYN attacks, described in Section V-C, in that the relative times to detection for different schemes, e.g. bPDM, MAD, PAD, are identical for both metrics. For example, the bPDM detects TCP SYN attacks faster than the other methods irrespective of the metric used for comparison. This equivalency exists for any anomalies that employ fixed-size packets. To compare the metrics’ efficacy in this case, we revisit Figure 7 in Section V-C as Figure 9, this time employing the packet SNR instead of the bitrate SNR. The times to detection for the three methods, averaged over 8 sets of synthetic TCP SYN attacks, are plotted as a function of packet SNR in Figure 9, with error bars representing the standard deviation of the detection times. In comparing Figures 7 and 9, we note that they are simply scaled and shifted versions of each other, which shows that the packet SNR is equivalent to the bitrate SNR in the case of attacks with *fixed-size packets*.

However, this is not always the case; we now consider an attack due to a *smart adversary*, as introduced in Section V-E. We compare Figure 10(a) to 10(b), wherein the bPDM times to detection are plotted for the smart and TCP SYN synthetic attacks as a function of bitrate SNR and packet SNR, respectively. Note that in Figure 10(b), the smart adversary attacks are detected more quickly than the corresponding TCP SYN attacks. This is a very counterintuitive result, since, as described above, the smart adversary represents a set of countermeasures against our detection mechanism. In contrast, in Figure 10(a), the TCP SYN attacks are shown to be detected markedly and uniformly faster than the smart adversary attacks for the entire range of bitrate SNR values, which is the result we expect.

Given that both metrics are equivalent in the case of attacks that use fixed-size attack packets, but that the packet SNR yields a counter-intuitive result in the case of the smart adversary, we conclude that the bitrate SNR is an effective metric for comparison and evaluation, and better than the packet SNR.

G. Validation of Synthetic Attacks

In order to confirm the conclusions drawn from the bPDM’s performance on simulated attacks, we here test the bPDM using three real network attacks captured in the wild and available through PREDICT, and three proxy-real attacks consisting of real denial-of-service attacks (DoS) and real background traffic streams combined using Stream Merger [22]. We find that the detection times for the real and proxy-real attacks closely resemble those of the synthetic attacks.

Though the three real network attacks considered were collected in varying network conditions, all six attacks employ

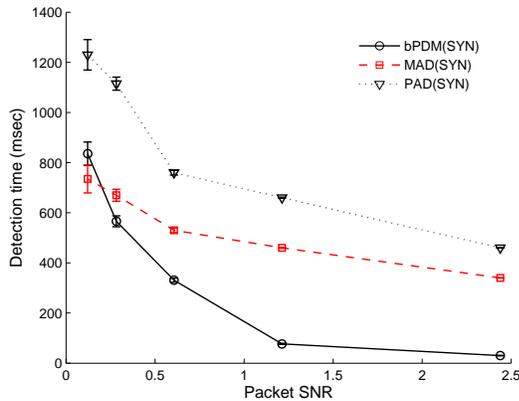


Fig. 9. Comparing the detection times (in msec) for the bPDM, MAD and PAD detection algorithms as a function of the packet SNR metric as defined in [17].

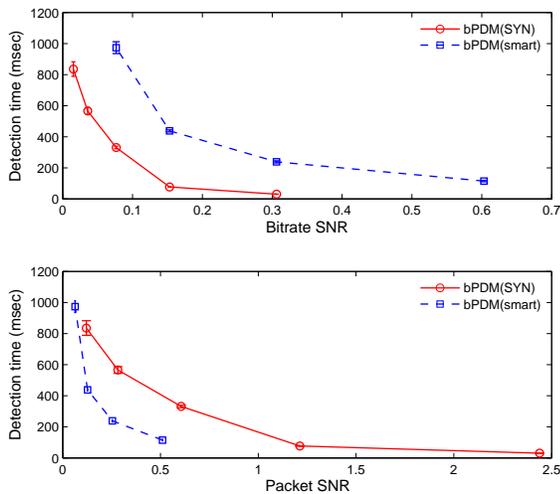


Fig. 10. Comparing the bPDM detection times for the set of synthetic TCP SYN and smart attacks as a function of (a) bitrate SNR and (b) packet SNR.

either 15-byte, 60-byte or 68-byte fixed-size attack packets. In this respect, they resemble the set of synthetic TCP SYN attacks. Thus, we expect the detection times for the real attacks to resemble those for the SYN attacks. Table IV summarizes the attack details; the bitrate SNRs for the real attacks range from 0.012 to 0.53, with varying attack and background traffic levels (in Mbps).

The detection times for the individual real and proxy-real attacks are plotted (as open, unconnected symbols) in Figure 11 for comparison to the detection times for the synthetic traces of the bPDM, MAD and PAD detection schemes. As in earlier plots, the points for the synthetic traces (*i.e.* bPDM(SYN), bPDM(smart), MAD(SYN) and PAD(SYN)) represent the mean detection times, with the error bars providing the standard deviation. The bPDM detection times for the real and proxy-real network attacks as shown in Figure 11 were obtained by running the algorithm on the attacks with 2–3 seconds of background traffic before the onset of the attack. Furthermore, the plot of the detection times for the synthetic

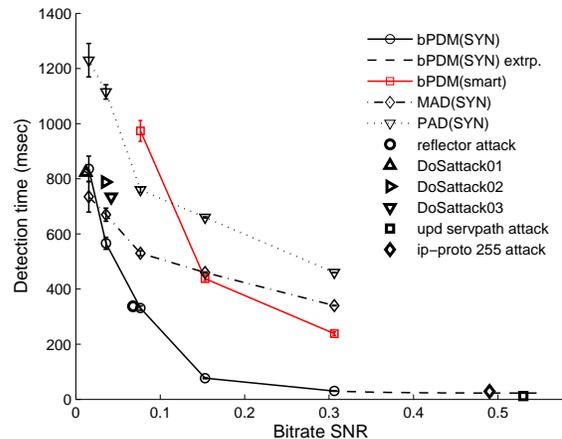


Fig. 11. Comparing the bPDM detection times (in msec) of the real network attacks to the bPDM, MAD and PAD detection times for the simulated synthetic attacks, TCP SYN and smart adversary.

TCP SYN attacks has been extrapolated (as indicated by a dashed black line) to compare the TCP SYN detection times to those for the real ip-proto 255 and UDP Servpath attacks. Of the models considered, the best fit for the synthetic TCP SYN detection times was found to be an exponential one, which is described as $f(\text{detection time}) = 22.6 + 1053e^{-16.08 \cdot \text{bitrate SNR}}$.

For all real and proxy-real attacks, we see that a higher bitrate SNR corresponds to a lower time to detection, and further note that the actual (and extrapolated) detection times for the synthetic TCP SYN attacks are consistent with the times to detection for the real and proxy-real network attacks.

H. Considering the Effect of Hop Count on Time to Detection

Recall that Figure 5 in Section V-A1 shows that the detection times for the synthetic TCP SYN attacks are consistently lower than the averaged Iperf detection times, described in Sections V-A1 and V-A2, respectively.

We believe this disparity is due to the shorter path for our synthetic TCP attacks: our synthetic TCP SYN attacks are constructed by mixing traffic using Stream Merger, so the attack traverses the equivalent of one hop. In contrast, the emulated Iperf attacks from CSU to USC must traverse the 8–10 hops over the real Internet (as measured by traceroute, see Appendix A), and at each hop through a router, cross-traffic from other applications can distort the attack. Although this conjecture is consistent with the data, its verification in a controlled experiment is an area for future work.

I. Controlling the Probability of False Positives

As discussed in Section III, the SPRT allows control over the probability of false positives via tunable parameters $\alpha = P_{FP}$ and $\beta = P_{FN}$, which directly affect the SPRT thresholds A and B . To examine the effect of these parameters on the probability of false positives, we employ a set of 24 background traffic traces [2], each 5 minutes long, gathered every hour for 24 hours.

TABLE IV
BPDM DETECTION RESULTS FOR REAL NETWORK ATTACKS.

Symbol	Trace	Attack (Mbps)	Background (Mbps)	bitrate SNR	TD (msec)	Description
◇	[4]	34.45	69.86	0.49	29	ip-proto 255 attack
○	[4]	2.11	31.12	0.678	338	reflector attack
□	[1]	21.6	40.75	0.53	12	udp servpath attack
△	[3]	3.84	320	0.012	823	proxy-real DoS attacks
▷	[3]	11.2	320	0.035	788	
▽	[3]	13.44	320	0.042	734	

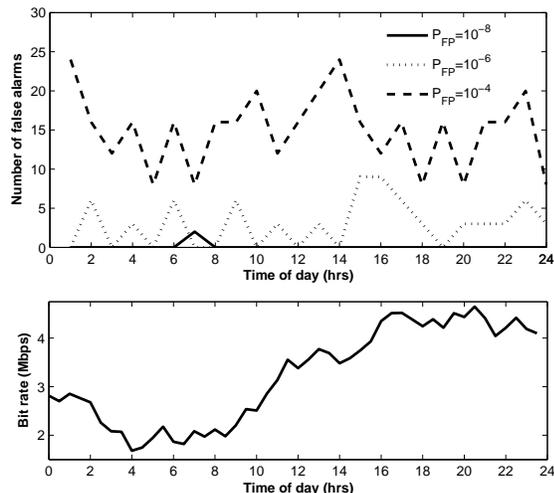


Fig. 12. (a) Number of false positives, and (b) the bitrate (in Mbps) as a function of time of day.

Figure 12 shows (a) the number of false positives declared by the bPDM for different values of P_{FP} , and (b) the bitrate (in Mbps) as a function of time of day for a 24-hour period. As expected, the number of false positives increases as P_{FP} increases, which is equivalent to lowering the upper threshold, $\log(B)$, of the SPRT. Furthermore, we see that the number of false positives declared by the bPDM is not correlated with the level of the background traffic, implying that the bPDM is fairly robust against variations in the background traffic. For the bPDM simulations using the simulated synthetic, emulated Iperf and real traces, we use $P_{FP} = 10^{-8}$, $P_{FN} = 10^{-7}$ and note that false positives are reduced to zero, which corresponds to the $P_{FP} = P_{FN} = 10^{-8}$ plot in Figure 12.

VI. CONCLUSIONS

We have developed the bivariate Parametric Detection Mechanism (bPDM), which can detect anomalies and low-rate attacks in a few seconds. This approach allows the real-time estimation of model parameters, and only requires 2–3 seconds of background-only traffic for training. Incorporating the packet-rate and packet-size features enables us to detect anomalies in encrypted traffic and avoid state-intensive flow tracking, since our method does not use flow-separated traffic; furthermore, combining these same two features also eliminates most false positives. We have evaluated our methods using synthetic traces and emulated Iperf attacks, and find that the bPDM can detect attacks in a few seconds. The detection

times for the synthetic attacks are validated using real and proxy-real network attacks, and the bitrate SNR is shown to be not only an effective metric for evaluating anomaly detection methods, but also a better one than the previously-proposed packet SNR metric. For all the datasets considered, as well as the underlying theoretical model, we find that the time to detection decreases as the bitrate SNR increases. Furthermore, we examine the effect of the individual components of the bitrate SNR on the time to detection and find that as the attack rate increases, the detection time decreases, and as background traffic level increases, the time to detection decreases.

APPENDIX A GENERATION OF IPERF ATTACKS

Our evaluation of the bPDM uses controlled Iperf attacks in varying Internet traffic mixes. These 80-second UDP attacks are generated at a fixed attack rate (in Mbps) and employ 345-byte fixed-size attack packets. The Iperf attacks originate at Colorado State University (USC) and are destined for the University of Southern California, with 10 routers traversed between the source and destination as determined by traceroute. The network packet traces consisting of these attacks are captured via port-mirroring by capture machines that use DAG cards and see both incoming and outgoing university traffic. In particular, we use one link (out of five) at Los Nettos, a regional ISP in the Los Angeles area serving both commercial and academic institutions. The traces are collected at Los Nettos with a timing precision of 0.1 microsecond, which is due to the accuracy of the Endace DAG network card. The link we use captures bidirectional traffic, but since the bPDM operates on a unidirectional traffic stream, the incoming traffic is filtered from the bidirectional traffic using a complete list of destination IP subnets for the University of Southern California. Once the incoming traffic has been isolated, the bPDM exploits only the aggregate traffic fields, the timestamp and the packet size, which yield the packet-rate and entropy of packet-size distribution statistics.

We collected four datasets, each 3 hours long, and consisting of an average of 15 Iperf attacks with attack rates of 20, 25, 30 and 40 Mbps. The experiments were conducted both during weekend non-peak hours and during busier weekday hours, to investigate the effect of different background traffic levels. We see qualitatively similar results across all the datasets, in that the time to detection is uncorrelated with the time at which the datasets were collected, and thus we do not analyze the dataset-partitioned data.

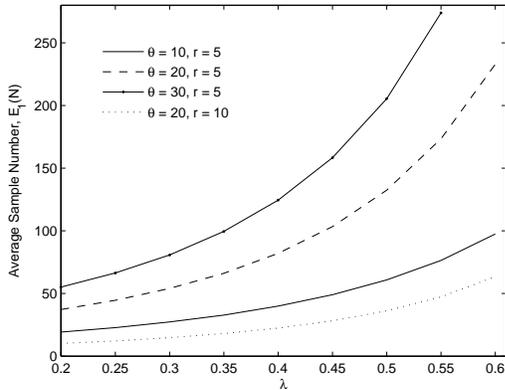


Fig. 13. For $\alpha = \beta = 10^{-5}$, ASN function $\mathbb{E}_1(N)$ computed for increasing λ and typical values of θ and r .

APPENDIX B AVERAGE SAMPLE NUMBER ANALYSIS

The average sample number (ASN) function is used to evaluate the efficacy of the sequential test reviewed in Section III. The ASN function is simply the average number of samples required to make a decision by a particular test. For the binary hypothesis test considered in our work, the ASN function is denoted $\mathbb{E}_i(N)$ for hypothesis H_i . We present an analysis of the ASN function for the GPD/sGPD hypothesis test since it is used to determine an alternative window size for parameter estimation. In order to compute the ASN function, we first define $z = \log \frac{p(x|H_1)}{p(x|H_0)}$, and denote $\mathbb{E}_\theta(z)$ to be the expected value $\mathbb{E}(z)$ of z when $\theta \in \{0, 1\}$ is the true hypothesis. From the expressions in (3), we can solve for

$$\alpha = \frac{A-1}{A-B} \quad \text{and} \quad \beta = -\frac{A(B-1)}{A-B}, \quad (23)$$

and now obtain expressions for the ASN functions for each of the hypotheses as [16]:

$$\mathbb{E}_0(N) = \frac{\alpha \log B + (1-\alpha) \log A}{\mathbb{E}_0(z)}, \quad (24)$$

$$\text{and } \mathbb{E}_1(N) = \frac{(1-\beta) \log B + \beta \log A}{\mathbb{E}_1(z)}. \quad (25)$$

Thus, for the GPD/sGPD hypothesis test, given the probability mass functions in (7) and (9), we derive

$$z = (x-r-1) \log[\theta + \lambda(x-r)] + \lambda r + \log[x!] - (x-1) \log[\theta + \lambda x] - \log[(x-r)!], \quad (26)$$

and then compute $\mathbb{E}_\theta(z)$ numerically. The ASN function for hypothesis H_1 is computed using (25) and is plotted in Figure 13 for typical values of θ and r for varying λ .

The sizes of the update windows are chosen to be on the order of the ASN function for the respective hypotheses to ensure that the parameter estimates, computed using the observations in those windows, correspond to a decision being made by the SPRT. When an anomaly is detected, the ASN functions and update window sizes are recomputed using the current estimate \hat{r} .

ACKNOWLEDGMENTS

Research has been funded by DHS NBCHC040137 and NSF CNS-0626696. Traces used in this work were provided by the USC/LANDER project.

REFERENCES

- [1] attack-servpath-udp22-20061106, available through www.predict.org.
- [2] Bottleneck_traces-20041202, available through www.predict.org.
- [3] DoS_80_timeseries-20020629, available through www.predict.org.
- [4] DoS_traces_20020629, available through www.predict.org.
- [5] UniformAttack_Traces_Generated20070821-20041202, available through www.predict.org.
- [6] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the SIGCOMM Internet Measurement Workshop*, France, November 2002.
- [7] M. Basseville and I. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [8] M. Brunner. *Service Provision: Technologies for Next Generation Communications*, chapter Basic Internet Technology in Support of Communication Services. Wiley Series on Communications Networking and Distributed Systems. Wiley, 2004.
- [9] Y. Chen and K. Hwang. Spectral Analysis of TCP Flows for Defense against Reduction-of-Quality Attacks. In *Proc. of the IEEE Intl. Conf. on Communications*, Glasgow, Scotland, June 2007.
- [10] P. Consul. *Generalized Poisson Distributions: Applications and Properties*. Marcel Dekker Inc., New York, NY, 1989.
- [11] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
- [12] H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.
- [13] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg. Rule-based anomaly detection on IP flows. In *Proceedings of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
- [14] J. Ellis and T. Speed. *The Internet Security Guidebook: From Planning to Deployment*. Academic Press, 2001.
- [15] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. Statistical approaches to DDoS attack detection and response. In *Proc. of DARPA Information Survivability Conf. and Exposition*, pages 303–314, 2003.
- [16] Z. Govindarajulu. *Sequential Statistics*. World Scientific Publishing, Singapore, 2004.
- [17] X. He, C. Papadopoulos, J. Heidemann, U. Mitra, and U. Riaz. Remote detection of bottleneck links using spectral and statistical methods. *Computer Networks*, 53:279–298, 2009.
- [18] A. Hussain, J. Heidemann, and C. Papadopoulos. Identification of repeated denial of service attacks. In *Proceedings of the Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006.
- [19] C. Jin, H. Wang, and K. Shin. Hop-count filtering: An effective defense against spoofed DoS traffic. In *Proceedings of Conference on Computer and Communications Security*, Washington DC, October 2003.
- [20] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *11th International WWW Conference*, Honolulu, HI, May 2002.
- [21] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
- [22] P. Kamath, K.-C. Lan, J. Heidemann, J. Bannister, and J. Touch. Generation of high bandwidth network traffic traces. In *Proceedings of the 10th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Fort Worth, TX, 2002.
- [23] S. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR, 1993.
- [24] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.
- [25] G. Nychis, V. Sekar, D. G. Andersen, et al. An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the 8th ACM SIGCOMM Internet Measurement Conference*, pages 151–156, Vouliagmeni, Greece, October 2008.
- [26] R. S. Prasad and C. Dovrolis. Beyond the model of persistent tcp flows: Open-loop vs closed-loop arrivals of non-persistent flows. In *Proceedings of the 41st Annual Simulation Symposium*, Ottawa, Canada, April 2008.

- [27] J. Rodríguez, A. Briones, and J. Nolasco. Dynamic DDoS mitigation based on TTL field using fuzzy logic. In *Proceedings of 17th International Conference on Electronics, Communications and Computers*, Cholula, Mexico, February 2007.
- [28] R. Sinha, C. Papadopoulos, and J. Heidemann. Internet packet size distributions: Some observations. Technical Report ISI-TR-2007-643, University of Southern California, Los Angeles, CA, USA, May 2007.
- [29] V. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. In *Proceedings of IEEE GLOBE-COM*, Dallas, TX, November 2004.
- [30] A. Soule, K. Salamatian, and N. Taft. Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 2005 Internet Measurement Conference*, Berkeley, CA, October 2005.
- [31] M. Stoecklin, J.-Y. L. Boudec, and A. Kind. A two-layered anomaly detection technique based on multi-model flow behavior models. *Lecture Notes in Computer Science (PAM)*, 4979:212–221, 2008.
- [32] B. Tabachnick and L. Fidell. *Using Multivariate Statistics (5th Edition)*. Allyn and Bacon, 2006.
- [33] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing*, 54(9):3372–3382, 2006.
- [34] G. Thatte, U. Mitra, and J. Heidemann. Detection of low-rate attacks in computer networks. In *Proceedings of IEEE 11th Global Internet Symposium*, Phoenix, AZ, April 2008.
- [35] G. Thatte, U. Mitra, and J. Heidemann. Parametric methods for anomaly detection in aggregate traffic (extended version). Technical Report ISI-TR-2009-663b, USC/Information Sciences Institute, Los Angeles, USA, August 2009.
- [36] M. Thottan and C. Ji. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, August 2003.
- [37] H. V. Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley, New York, 1968.
- [38] A. Wagner and B. Plattner. Entropy based worm and anomaly detection in fast IP networks. In *Proceedings of the SIG SIDAR Graduierten-Workshop uber Reaktive Sicherheit*, Berlin, Germany, July 2006.
- [39] A. Wald. *Sequential Analysis*. John Wiley, New York, 1947.
- [40] H. Wang, D. Zhang, and K. Shin. Detecting SYN flooding attacks. In *Proceedings of the Conference on Computer Communications (INFO-COM)*, New York, NY, June 2002.

John Heidemann is a senior project leader at USC/ISI and a research associate professor at USC in the CS Department. At ISI he leads I-LENS, the ISI Laboratory for Embedded Networked Sensor Experimentation, and investigates network protocols and Internet traffic as part of the ANT (Analysis of Network Traffic) lab. He received his B.S. from University of Nebraska-Lincoln and his M.S. and Ph.D. from UCLA, and is a senior member of ACM and IEEE.

Gautam Thatte is currently working toward the Ph.D. degree in electrical engineering at the University of Southern California (USC), Los Angeles. His current research interests are in the areas of optimization, estimation and detection in sensor networks and computer networks. He received his B.S. degree (distinction) in engineering from Harvey Mudd College (HMC), Claremont, CA, in 2003 and his M.S. degree in electrical engineering from USC, in 2004. He was awarded the USC Viterbi School of Engineering Deans Fellowship and the HMC International Student Scholarship, in academic years 2003-2004 and 1999-2003, respectively, and is a student member of the IEEE.

Urbashi Mitra received the B.S. and the M.S. degrees from the University of California at Berkeley in 1987 and 1989 respectively, both in Electrical Engineering and Computer Science. From 1989 until 1990 she worked as a Member of Technical Staff at Bellcore in Red Bank, NJ. In 1994, she received her Ph.D. from Princeton University in Electrical Engineering. From 1994 to 2000, Dr. Mitra was a member of the faculty of the Department of Electrical Engineering at The Ohio State University, Columbus, Ohio. In 2001, she joined the Department of Electrical Engineering at the University of Southern California, Los Angeles, where she is currently a Professor. Dr. Mitra is a Fellow of the IEEE.