A Decision-Theoretic Approach to Detect Anomalies in Internet Paths

Fida Hussain*, Umar Kalim†, Noman Latif* and Syed Ali Khayam*
*School of Electrical Engineering and Computer Sciences (SEECS)
National University of Sciences and Technology (NUST), Pakistan.
Email: {fida.hussain, 45noman, khayam}@niit.edu.pk
†Stanford Linear Accelerator Center (SLAC), Stanford University, CA, USA. 94025.
Email: kalim@slac.stanford.edu

***Abstract*—In this paper, we propose an algorithm that detects significant events on an Internet path by monitoring the available bandwidth. Evaluating a comprehensive dataset of diverse bandwidth measurements reveals that significant noisy traffic spikes are generally observed on Internet paths. To extract normal path characteristics from these noisy real-time measurements, we low-pass filter the bandwidth estimates and show that the distribution of normal path bandwidths approaches Gaussianity irrespective of the path being monitored. This Gaussian baseline model is then leveraged in a decision-theoretic framework to detect path events. We show that the proposed detector provides highly accurate performance and easily surpasses the accuracy of existing techniques.**

I. Introduction

Over the last decade, enterprise, academic and research networks have scaled dramatically in terms of their capacities, sizes, supported applications and services. Identification of anomalous events in these networks is becoming increasingly challenging for network operators as the anomalies have now become quite diverse. While anomaly detection in aggregate enterprise-level network traffic has received significant research attention in the last decade [1]–[5], detection of anomalous events on an *end-to-end* Internet path is largely unexplored. Detection of anomalous events on Internet connectivity paths between networks and regions is important because such event detection: 1) facilitates *network operations* [6] as it helps identify and quantify network path changes and provides alerts and diagnosis about whether the faults lie with the path or the applications; 2) allows *network planning* [7] by providing achievable performance and by maintaining historical information on network growth; and 3) provides *better insight* into the impact network performance on applications[1] and protocols.

To accurately classify interesting events, a path anomaly detection algorithm should gather sufficient statistical information before classifying an event as an anomaly. Such a strategy, however, results in undesirable detection, diagnosis and reaction delays. Thus an inherent tradeoff exists between detection delay and classification accuracy. In this paper, we first evaluate three existing path anomaly detectors: 1) the plateau algorithm by Logg et al. [12]; 2) the Kalman filter (KF) based detector by Augustin et al. [13]; and 3) the adaptive fault detector by Hajji [14]. Here we concern ourselves with end-to-end anomaly detection in terms of non-malicious events such as equipment failures (e.g., end-host failure, link outage), and uncharacteristic usage (e.g., flash crowds, high volume flows) and behavior (e.g., misconfigurations, fluttering in traffic routes). We use a comprehensive dataset [6] of bandwidth measurements collected over several geographically diverse paths using different tools (iperf [15], pathChirp [16], and thrulay [17]) for a period of up to three years. We label this dataset using a simple and unbiased labeling algorithm and show that the performances (in terms of accuracy of change detection and detection delay) of existing anomaly detectors have significant room for improvement. From the performance results of existing detectors, we note that a path anomaly detector should incorporate and leverage the inherent statistical characteristics of *normal* bandwidth measurements observed on Internet paths.

To extract normal path characteristics from aggregate real-time data, we remove the noisy bandwidth spikes by applying a low-pass median filter to the observed measurements. We show that the distribution of normal path bandwidths approaches Gaussianity when the bandwidth is measured using packet-pair dispersion. We leverage this baseline Gaussianity model of normal bandwidth measurements in a decision-theoretic framework to detect anomalous events on an end-to-end Internet path. Receiver operating characteristics (ROC) curves[2] and detection delay are used to evaluate the accuracy and timeliness of the proposed detector. We show that the proposed detector provides high accuracy of change detection with low detection delay and easily surpasses the performance of existing techniques.

This work is supported in part by the Higher Education Commission (HEC) of Pakistan under the project *Network Monitoring for PERN* (44-DDWP-2007) and the Mathematical and Computational Sciences Division under the U.S. Department of Energy. SLAC is operated by Stanford University for the U.S. Department of Energy under contract DE-AC02-76SF00515.

[1] These applications may vary from sophisticated software providing remote access to scientific instrumentation [8], [9] to adaptive protocols [10] and applications [11].

[2] The data sets, the source codes and the methods of obtaining the results are documented at https://confluence.slac.stanford.edu/display/IEPM/Decision+Theoretic+Approach

The rest of this paper is structured as follows. Section II describes the Internet datasets used to test, validate and compare anomaly detectors. Section III provides comparative analysis of existing anomaly detectors. Section IV outlines the proposed decision-theoretic detection method and compares its performance with existing methods. Section V summarizes key findings of this paper.

II. Data Collection and Labeling

To evaluate different Internet path anomaly detection algorithms, we use real world performance measurements collected by the Internet End-to-end Performance Monitoring Bandwidth (IEPM-BW) project [6]. The purpose of the IEPM-BW project is to develop an infrastructure based on standard open technologies to make active end-to-end application and network performance measurements and predictions. The measurements and predictions are targeted at high performance network links, such as those used worldwide by Grid applications and other A&R applications deployed over high performance networks such as ESnet, Internet2 and other A&R networks in the developed world. However, it does provide low impact network performance measurements to most of the Internet connected world providing delays, loss and connectivity information over long (several years) durations. In this section, we describe the IEPM measurement setup and then discuss preliminary statistics and labeling of the data.

A. IEPM Bandwidth Measurement Dataset

The first step towards Internet path anomaly detection is the identification of an end-to-end performance metric that is expected to exhibit sustained fluctuations during the course of an anomalous event. We note that among the available performance metrics (e.g., latency, jitter, packet-loss, number of connections per host, etc.,) *available path bandwidth* satisfies this requirement and hence is being used for event detection on Internet paths [12][3]. In this section, we describe and label the IEPM bandwidth measurement dataset used in this study.

1) IEPM Topology: The test sites of the IEPM-BW project include geographically diverse Academic and Research (A&R) institutes situated in Canada, Czech Republic, United Kingdom, France, Germany, Italy, Japan, Netherlands, Pakistan, Russia, Switzerland, Taiwan and USA. Details of the network topology can be obtained from [6].

For the purpose of this study, we selected six Internet paths which are between SLAC and a) San Diego Supercomputing Center (SDSC) USA, b) European Organization for Nuclear Research (CERN) Geneva, Switzerland, c) Forschungszentrum Karlsruhe (FZK) Germany, d) Deutsches Elektronen-Synchrotron (DESY) Germany, e) Oak Ridge National Laboratory (ORNL) USA and f) University of Toronto (UTORONTO) Canada. The reasons behind selecting these sites are three-fold: a) These sites use all the three performance measurements tools (iperf, pathChirp and thrulay) unlike others which deploy one or two of the three tools of our interest; b) They feature minimum downtime and hence do not suffer from large durations of missing data; and c) They span international boundaries.

TABLE I
Performance measurement tools.

Tool	Metric(s)
Ping [20]	Delay and loss
OWAMP [21]	One-way delay and one-way loss
Traceroute [20]	Path
IPerf [15]	Achievable throughput
pathChirp [16]	Available bandwidth
Thrulay [17]	Achievable throughput
Pathload [22]	Available bandwidth

2) IEPM Measurements: Table I lists the performance metrics observed and the tools used by the IEPM-BW project. Measurements are calculated in terms of minimum, average and maximum estimates from a series of tests scheduled every 30-45 minutes resulting in approximately 50 observations per day. From the available metrics, and based on the metrics used by prior studies [6], we chose the average available bandwidth estimate as input for the event detection algorithm because it is perturbed throughout the course of an anomaly. These features are outlined in Fig. 1. These variations are either significantly different from the normal behavior and/or persist for a noticeable duration.

B. Data Labeling Algorithm

The accuracy of an Internet path anomaly detector can only be evaluated on labeled data with clearly demarcated anomalous time periods. Since such labeling is not available for known end-to-end performance measurement datasets. Therefore, in this section we develop a simple and unbiased labeling algorithm.

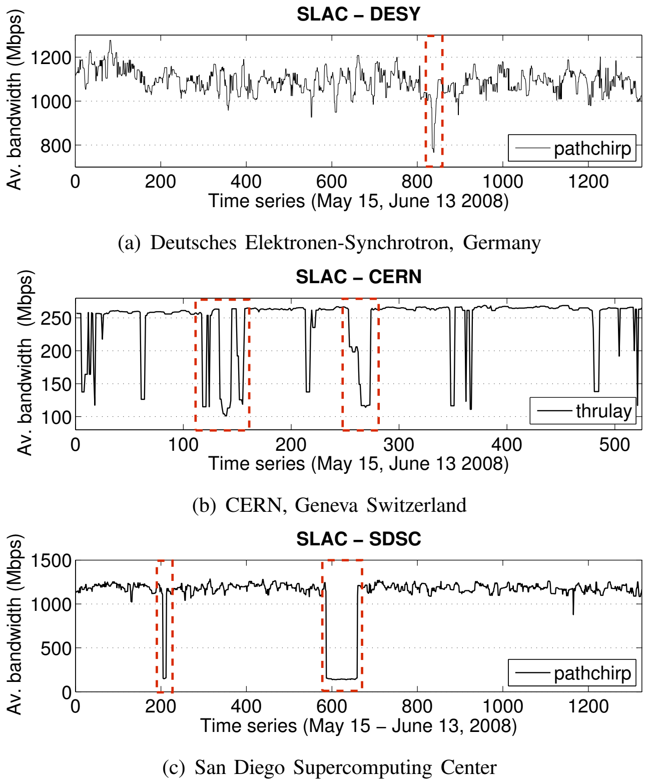
Before we describe the data labeling algorithm, we observe that an accurate labeling algorithm for the present problem should cater for the baseline or normal behavior of the bandwidth measurements. Moreover, this baseline behavior, whether it represents normal or anomalous behavior, should tend to sustain itself over (at least) a minimum defined duration. Based on these observations, we define an interesting Internet path event, in short an *event*, as:

Definition 1. A set of anomalous observations is called an *event* if the deviant observations persist for a period greater than or equal to a defined epoch δ.

The minimum event duration that was observed in the datasets under consideration was $\delta = 3$ hours.

Since a set of given bandwidth measurements contains both normal and anomalous observations, data labeling in the present context can be sub-divided into two steps: 1) Extraction of baseline or normal bandwidth values from a given set of noisy bandwidth measurements; 2) Identifying and demarcating anomalous bandwidth measurements that deviate

[3] With reference to the discussion in Section IV-A and [18], [19] we acknowledge that bandwidth measurements are not always accurate. Nevertheless, as long as the measurements exhibit sustained fluctuations during anomalous periods, measurement accuracy is not fundamentally important for the present problem.



(a) Deutsches Elektronen-Synchrotron, Germany

(b) CERN, Geneva Switzerland

(c) San Diego Supercomputing Center

Fig. 1. Samples of available bandwidth measurements as seen from SLAC with annotated anomalies.

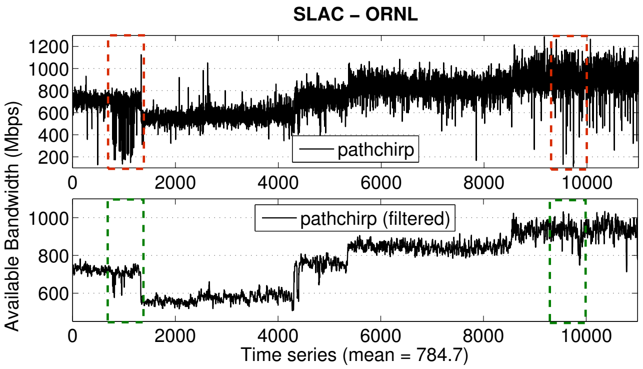


Fig. 2. Low-pass median filtering of bandwidth measurements to extract baseline behavior; the time series is annotated to show how median filtering results in removal of sustained anomalies and spurious measurements.

significantly from the baseline values. The following two sections elaborate on these steps.

1) Extracting Baseline (Normal) Behavior of Bandwidth Measurements: Given a set of bandwidth measurements, extraction of the baseline behavior essentially entails removing all anomalous observations (and the corresponding bandwidth values) from the set. The remaining measurements can then be used to characterize the baseline behavior. Anomalous bandwidth values always cause significant fluctuations in the measurements, albeit these fluctuations may be sustained or spurious in nature. These two types of anomalies are shown in Fig. 1. Both of these anomalies should be removed from the dataset before baseline behavior is characterized.

To remove the anomalous bandwidth measurements from the dataset, we apply an n-tap median filter to the dataset. A median filter is a sliding window low-pass filter that stores n previous values of the input and at each step outputs the median of the stored values. Consequently, high frequency spikes are removed from the input data. Note that the value of n is a crude upper bound on the maximum duration for an anomaly. If a bandwidth change sustains itself beyond n observations then it is treated as a change in the underlying baseline behavior. Therefore, care should be exercised in choosing the value of n for a given bandwidth measurement dataset. We define an empirical lower bound on n as:

$$n \geq 2\delta\upsilon,$$

where υ is the average number of IEPM performance measurements made in one hour. In the present dataset, we observed that a maximum value of $n = 15$ is sufficient to remove sustained and spurious bandwidth fluctuations. An example of the baseline (normal) bandwidth values extracted through median filtering is shown in Fig. 2.

2) Event Identification, Labeling and Demarcation: The median filtered dataset are treated as normal bandwidth values of an Internet path. The baseline behavior is then characterized by computing the mean of the filtered measurements. Then as per Definition 1, an event is flagged when an anomaly deviates significantly from this baseline behavior and sustains itself for more than $\delta = 3$ hours.

To flag significant deviations, we first compute the mean μ_f of the baseline. We then analyze the measurements in subsets (windows) of length δ. The mean μ_Δ of the window is computed and a test is performed such that:

$$0.5 \leq \frac{\mu_\Delta}{\mu_f} \leq 1.5. \tag{1}$$

We opt for such thresholds in light of Definition 1; empirical observation suggests that nearly 6% of the observations show a difference of greater than or equal to 50% from the mean observation[4] Also, such deviant observations tend to maintain their state and feature small variation (irrespective of the duration of the event) as shown in Figs. 1 and 2, thereby endorsing the fact that significant change is primarily observed in the mean observations (i.e. μ_Δ and μ_f) and not in the variance.

Once all observations are scrutinized, windows marked as anomalous are analyzed and coalesced[5] to identify the demarcations of unique events. The detailed data labeling procedure is described in Algorithm 1.

3) Discussion: The events labeled using Algorithm 1 were subsequently verified manually to ensure the correctness of the labeling. Using the labeled datasets, we evaluate the

[4] Details are available at https://confluence.slac.stanford.edu/display/IEPM/Decision+Theoretic+Approach

[5] Note that each unique event must be of a duration greater than δ. Also the separation between events must be greater than δ to classify the events as unique.

Algorithm 1: Labeling data with anomalies.

```
Data: a) Array of performance measurements $\Omega$, length of
      sliding window $\Delta$, the duration $\delta$ for which an abnormal
      activity needs to persist before it is considered an event
      and the average number $\upsilon$ of performance measurements
      per hour available in the dataset
Result: $\tau$: Array of time brackets defining all independent
        events
Compute $n = 2\delta\upsilon$;
Apply $n$-tap median filter to $\Omega$ to obtain $\Omega_f$ and consequently
$\mu_f$;
for $\{\omega_i \in \Omega | 1 \leq i \leq N\}$ do
    Compute $\mu_\Delta$ ;
    if $\left(0.5 \leq \frac{\mu_\Delta}{\mu_f} \leq 1.5\right)$ then
        Mark as normal observation;
    else Mark as an anomalous window and add to $\tau$;
end
for {all alerts in $\tau$}; do
    If required coalesce alert-windows considering $\delta$ to identify
    unique observations with adjusted boundaries;
end
```

performance of existing anomaly detectors in the next section. Before proceeding, we highlight that while the above data labeling algorithm is quite accurate, it cannot be used as an effective anomaly detector because it requires all bandwidth measurements to be available before the algorithm can start event classification. Consequently, while this algorithm can be used for offline data processing and labeling, it cannot be used for real-time event detection.

III. Comparison of Existing Path Anomaly Detectors

Performance of an Internet path event detector is defined by its accuracy (detection and false alarm rates) and the speed of event detection. More specifically, network traffic typically shows three types of variations [23]: 1) daily periodic behavior or diurnal patterns, 2) random and sporadic fluctuations, and 3) occasional bursts of high or low network activity. Since the first two types of variations do not warrant remedial measures, they are not *interesting* for network operators. The third type of traffic variation satisfies our definition of an event (Definition 1) as it causes prolonged perturbations in an end-to-end path and therefore requires immediate attention. The problem then is: When does an event being treated as uninteresting (diurnal or sporadic) become interesting? An inherent tradeoff between accuracy and delay can be observed here. If we wait long enough for more measurements to arrive before flagging the current measurements as anomalous, the accuracy in detecting interesting events will improve. However, such a procedure will lead to significant detection delays which are highly undesirable in the present problem. A good path anomaly detector should balance this accuracy-delay tradeoff.

Based on the above discussion, in this section we compare the accuracies and detection delays of the following three existing detectors: 1) the plateau algorithm by Logg et al. [12]; 2) the Kalman filter (KF) based detector by Augustin et al. [13]; and 3) the adaptive fault detector by Hajji [14]. To maintain a logical flow of thought, we briefly describe these algorithms in the following section. The rest of this section provides detailed performance evaluation of these anomaly detectors on the labeled IEPM dataset.

A. Description of Existing Algorithms

1) Plateau Algorithm [12]: The plateau algorithm [12] is the currently-deployed change detection algorithm that monitors bandwidth to the sites in IEPM-BW project [6], [8]. The algorithm, which evolved from [24], flags significant deviations in the mean and standard deviation of real-time bandwidth observations. Flagged measurements are compared against user-defined thresholds and classified as normal or anomalous.

2) Kalman Filter (KF) based Detector [13]: Augustin et al. [13] filter out the characteristic behavior of an Internet path using Kalman filters. The residuals are then investigated for potential anomalies. Four different methods are used: 1) to compare the residuals to a user-defined threshold; 2) as an extension of [3], to compare the local variance to the global variance assessment; 3) To apply wavelet analysis on the filtered data, unlike [3] which does the same for raw data; and 4) to define a likelihood ratio test to identify change in the mean rate of the residual signal. It is observed that the wavelet analysis performs poorly in comparison to the user defined threshold as well as the likelihood ratio test as the intuition that *an anomaly should diffuse itself at several time scales* is not realized as such.

3) Adaptive Fault Detection (AFD) Method [14]: Hajji [14] models traffic measurements as a K-variate Gaussian distribution. The model operates on an increment process that observes differences between consecutive values rather than operating on the original measurements. The detection procedure includes two phases: 1) training a baseline model for the network traffic increments using the expectation-maximization algorithm; and 2) flagging sudden changes using the likelihood ratio test.

B. Evaluation of the Existing Detectors

1) ROC Curves: We compare the accuracies of the detectors through ROC curves [25]. ROCs are commonly used to evaluate the performance of classification algorithms. These curves are used extensively in signal processing, intrusion detection, medicine, machine learning and data-mining communities [25]. ROC curves are well-suited for performance evaluation of classification algorithms because they organize the performance of an observed algorithm for the complete range of its tuning parameters (or threshold settings). Before proceeding to the evaluation, we define the main performance evaluation metrics used in ROC curves of the present problem:

Definition 2. A *true-positive* is the correct classification of an anomalous bandwidth event.

Definition 3. A *false-positive* is the incorrect classification of a normal bandwidth measurement as anomalous.

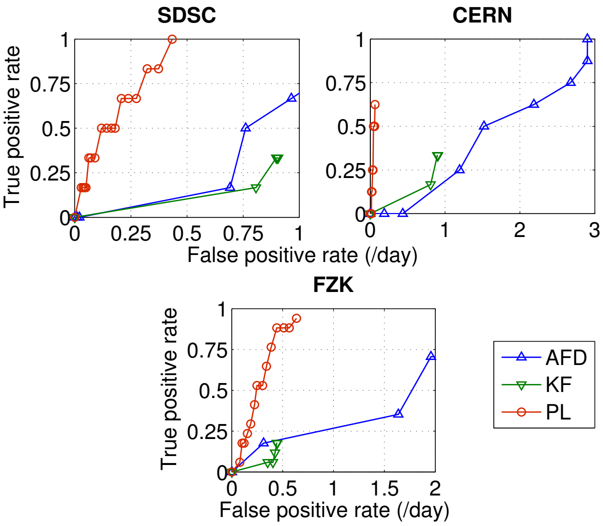


Fig. 3. ROC curves of Kalman Filters method (KF), Adaptive Fault Detection (AFD) and the Plateau Algorithm (PL) for pathChirp measurements as seen from SLAC.

Definition 4. *True-positive rate* is the ratio of the correctly classified events to the total number of events present in a dataset.

Definition 5. *False-positive rate* is defined as the ratio of the incorrectly classified normal values to the total number of days observed.

We draw the ROC curves with the *true-positive rate* on the Y-axis and the *false-positive rate* on the X-axis. Each point on the ROC curve represents performance results for one configuration (or threshold value) whereas the curve represents the behavior for the complete set of configurations. When compared, the steepest curve is considered the best as it approaches the highest true-positive rate with the lowest false-positive rate.

2) Comparison of Change Detection Accuracies: To generate ROC curves, we varied the buffer lengths and the threshold values of the algorithms. The ROC curves of the AFD, KF and PL algorithms are shown in Fig. 3. We observe that both the AFD and KF perform poorly with unacceptable levels of false-positive rates for desired change-detection rates. When applied to the Internet path SLAC-CERN, the AFD method at best results in a true-positive rate of 1 and correspondingly 3 false-positives per day. Similarly, as shown in Table II, the AFD method achieves 100% true-positive rate with a corresponding false positive rate of 7.07 for the Internet path SLAC-UTORONTO. The plateau algorithm, on the other hand, performs much better with a true-positive rate of approximately 0.75 against a relatively low false-positive rate of 0.5 incorrect alarms per day.

TABLE II
PERFORMANCE RESULTS FOR THE AFD ALGORITHM (SLAC TO UTORONTO) (TRUE ANOMALIES = 38, NUMBER OF DAYS = 900).

Desired detection rate	Desired false positive rate	Number of true positives	Number of false positives	True positive rate	False positive rate
0.95	0.02	1	19	0.03	0.02
0.95	0.01	4	291	0.11	0.32
0.95	0.009	4	438	0.11	0.49
0.95	0.007	6	764	0.16	0.85
0.95	0.006	14	1057	0.37	1.17
0.95	0.003	24	2199	0.47	2.44
0.95	0.002	28	4457	0.74	4.95
0.95	0.0018	35	4806	0.92	5.34
0.95	0.001	38	6361	1.00	7.07

AFD has poor accuracy because it relies on the assumption that the difference between consecutive normal measurements is small. While this assumption holds for frequent bandwidth measurements, in case of measurements that are spread out in time (e.g., the IEPM-BW measurements every 30-45 mins,) large variations between consecutive measurements enhance the sensitivity and the false-positive rate of the AFD algorithm. Kalman filter fails primarily because it assumes that bandwidth measurements are corrupted by an additive Guassian noise process, an assumption that does not hold in the present context. Plateau provides better accuracy because, instead of making assumptions about the bandwidth or noise processes, it leverages the mean and standard deviation of the real-time bandwidth measurements.

3) Delay Comparison: Detection delay is generally defined as the time taken by an anomaly detector in identifying an anomalous event. Since IEPM-BW's measurements are made with regular intervals, we define detection delay as the difference between the first observation flagged as anomalous by an algorithm and the first actual anomalous observation of the event.

Fair comparison of detection delays is difficult because different detectors feature different false positive and detection rates. Consider, for instance, an anomaly detector that classifies all bandwidth measurements as anomalous. Now while this detector is a completely impractical and the most inaccurate detector, its detection delay will be zero. Therefore, fair comparison of detection delays requires that delay is computed for a practical point on the ROC curve. To this end, for each detector we select the ROC point of the detector having the maximum possible detection rate; PL in the present case. For the highest detection rate PL detector, the detection delays of all detected events are computed. For the remaining detectors, we compute detection delays at ROC points having similar false alarm rate as the PL detector. Average detection delays of all detectors are computed by simply adding the delay in detecting each event divided by the number of detected events. Also we define the detection delays of events not detected by an anomaly detector as ∞.

Delay results for the Internet paths between SLAC and UTORONTO, CERN, DESY, SDSC and FZK are listed in Table III. Plateau and the Kalman Filter method provide similar detection delays, while the Adaptive Fault Detection method requires a significantly larger number of observations before an event is detected. We also observed with Plateau

TABLE III
AVERAGE DETECTION DELAY $\bar{\varepsilon}$ (IN TERMS OF ADDITIONAL OBSERVATIONS REQUIRED BEFORE AN EVENT IS DETECTED).

	Plateau					
	Detected		*Undetected*		*Total*	*False positives*
	#	$\bar{\varepsilon}$	#	$\bar{\varepsilon}$		
UTOR	23	4.98	15	∞	38	217
CERN	1	2.00	7	∞	8	19
DESY	14	12.60	17	∞	31	97
SDSC	1	0.00	5	∞	6	14
FZK	7	28.53	10	∞	17	117
	Adaptive Fault Detection					
	Detected		*Undetected*		*Total*	*False positives*
	#	$\bar{\varepsilon}$	#	$\bar{\varepsilon}$		
UTOR	4	53.25	34	∞	38	219
CERN	0	0.00	8	∞	8	193
DESY	1	47.43	30	∞	31	31
SDSC	0	0.00	6	∞	6	5
FZK	3	110.53	14	∞	17	163
	Kalman Filters					
	Detected		*Undetected*		*Total*	*False positives*
	#	$\bar{\varepsilon}$	#	$\bar{\varepsilon}$		
UTOR	4	4.75	34	∞	4	227
CERN	0	0.00	8	∞	0	819
DESY	0	0.00	31	∞	1	309
SDSC	0	0.00	6	∞	0	-
FZK	0	0.00	17	∞	3	-

TABLE IV
GOODNESS-OF-FIT TEST RESULTS FOR MEASUREMENTS (AS SEEN FROM SLAC) FITTING A GAUSSIAN DISTRIBUTION.

Site	χ^2	p-value	C.Val@0.05	C.Val@0.001
UTOR	14.18	0.2892	21.026	32.909
CERN	15.46	0.2170	21.026	32.909
DESY	17.30	0.1385	21.026	32.909
ORNL	18.22	0.1089	21.026	32.909
FZK	19.09	0.0864	21.026	32.909
SDSC	21.02	0.1009	23.685	36.123

that reducing the size of the buffers results in a decrease in the detection delay but has an adverse effect of making the algorithm sensitive to spurious changes which subsequently increase the algorithm's false positive rate.

C. Discussion

Based on the results of this section, the accuracies of existing anomaly detectors leave significant room for improvement. Overall, we observed that all of the existing anomaly detectors are general-purpose anomaly detectors which are designed to flag changes in any underlying observation metric. We show in the following section that bandwidth measurements on Internet paths exhibit some very specific characteristics that can facilitate classification. However, existing algorithms do not take these inherent bandwidth characteristics into account and are therefore unable to provide the required performance. The next section reveals useful statistical characteristics of Internet path bandwidth measurements and then proposes a anomaly detector that leverages these characteristics in a decision-theoretic framework.

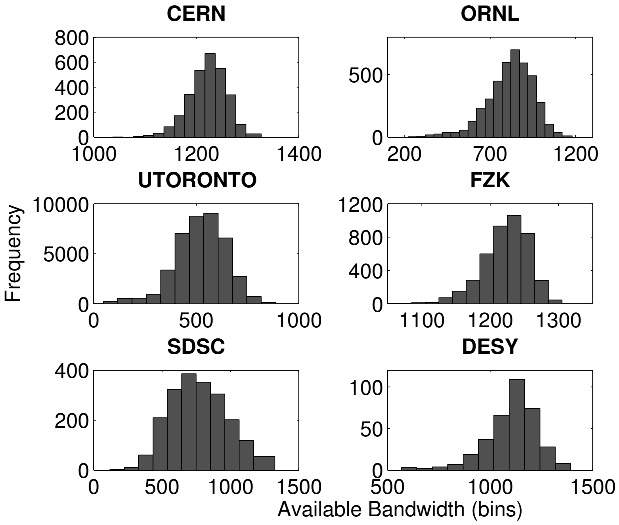


Fig. 4. Histograms of randomly selected pathChirp bandwidth measurement samples (as seen from SLAC).

IV. BANDWIDTH STATISTICS AND THE DECISION-THEORETIC ANOMALY DETECTOR

In this section, we first show that the baseline behavior of path bandwidth measurements collected using packet-pair dispersion techniques exhibit Gaussianity. Observations that deviate from the Gaussian behavioral model can thus be classified as anomalous. We then use the Gaussian baseline model in a decision-theoretic framework for real-time anomaly detection.

A. Statistical Behavior of Available Bandwidth Measurements

We randomly selected sample subsets from the IEPM data to identify the baseline characteristics of the observed Internet paths. These sample subsets included observations made over three or more consecutive days. A window of three days was selected because we observed in the IEPM dataset that anomalies generally persist for less than three days and conversely any change persisting beyond three days tends to be permanent.

In more than two-thirds of the pathChirp subsets, we observed that the distribution of bandwidth measurements approach Gaussinity. Examples of these Gaussian subsets are shown in Table IV and Fig. 4. This is an important statistical characteristic of the underlying normal pathChirp bandwidth measurements which can and should be leveraged for baseline behavior characterization and subsequently for anomaly detection. We use this baseline Gaussian behavior of pathChirp measurements in a decision-theoretic anomaly detection framework in the next section. Before we proceed further, a note on the bandwidth distributions of the other two tools (iperf and thrulay) is in place. Assuming that the three performance measurement tools were configured and deployed correctly, one would expect that the bandwidth estimates and their statistical properties provided by different tools for the same path would at least track each other if

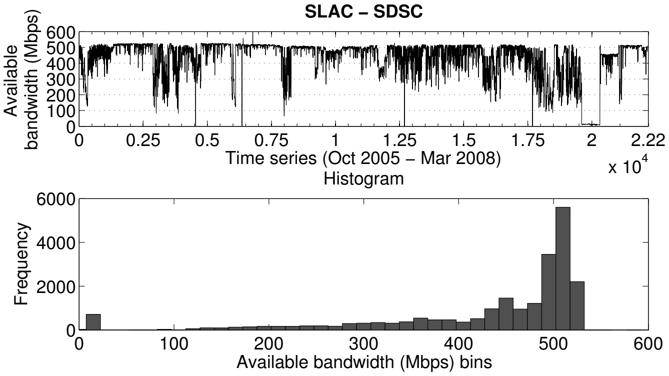


Fig. 5. Measurements made by thrulay from SLAC to SDSC.

not be exactly identical. However, comparison of the available bandwidth estimates by pathChirp, iperf and thrulay showed clear inconsistencies. As an example, thrulay's bandwidth estimates and their resultant frequency distribution for the SLAC-SDSC link are shown in Fig. 5. Clearly, this distribution is very different from a Gaussian distribution. In particular, we observed that bandwidths measured by thrulay and iperf for high-speed links were similar, while the estimates made by pathChirp were significantly greater than those of thrulay and iperf. Further investigation revealed that the factor influencing the measurement results for iperf and thrulay was the underline TCP stack; the same conclusion was reached in an independent study of [26] which came to our attention much later in this effort. For IEPM-BW the underline TCP stack is TCP New Reno[6] [27]. It has been shown earlier [28], [29] that Reno's Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm works well for low-speed links, but AI is very slow and MD is too drastic for high-speed links. This discrepancy results in poor link utilization by thrulay and iperf, and consequently their bandwidth estimates are different from the low-throughput pathChirp tool. Based on these results and [18], [19] we acknowledge that bandwidth measurements are not always accurate. However, as long as the measurements feature significant and sustained perturbations during anomalous periods, measurement accuracy is not fundamentally required for change detection. Thus, while considering the discussion in Section II-B, we advocate the use of pathChirp as an available bandwidth measurement tool for anomaly (particularly change) detection over Internet paths links. Henceforth, we only report results using the pathChirp datasets. The following section develops the decision-theoretic model to detect anomalies in the bandwidth measurements.

B. Decision-Theoretic Model of Bandwidth Measurements

Let $\mathcal{R}_i$ be the bandwidth measurements by pathChirp. These measurements either reflect the baseline or normal behavior of the path (i.e., the internal response [30]) or the anomalous observations (i.e., the internal response modified by noise). We define two hypotheses: $\mathcal{H}_0$, the null hypothesis where $\mathcal{R}_i$ represents the internal response (i.e., the baseline characteristics); and $\mathcal{H}_1$, the alternate hypothesis where R_i represents the internal response modified by noise (i.e., anomalous activity). This can be summarized as follows:

$$\mathcal{H}_0 : \mathcal{R}_i = n \tag{2}$$

$$\mathcal{H}_1 : \mathcal{R}_i = n + m_i, \tag{3}$$

where n represents a Gaussian random variable characterizing the baseline distribution of bandwidth estimates. For ease of exposition, we remove the first moment bias from n to make it a standard normal distribution $\mathcal{N}(0, \sigma^2)$. Also, we may represent m_i as: $m_i = \mathcal{R}_i - n$.

When a new bandwidth estimate arrives, it is mapped to one of the two hypotheses using the following conditional probability distributions:

$$\begin{aligned} Pr(\mathcal{R}_i|\mathcal{H}_0) &= \tfrac{1}{\sigma\sqrt{2\pi}} \exp\left(\tfrac{-\mathcal{R}_i^2}{2\sigma^2}\right); \text{ and} \\ Pr(\mathcal{R}_i|\mathcal{H}_1) &= \tfrac{1}{\sigma\sqrt{2\pi}} \exp\left(\tfrac{-(\mathcal{R}_i - m_i)^2}{2\sigma^2}\right). \end{aligned} \tag{4}$$

A likelihood ratio test [31] to choose between the two hypotheses can then be defined as:

$$\Lambda(\mathcal{R}_i) = \frac{Pr(\mathcal{R}_i|\mathcal{H}_1)}{Pr(\mathcal{R}_i|\mathcal{H}_0)}. \tag{5}$$

Assuming independence between real-time bandwidth measurement, an aggregate likelihood for a set of measurements $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_N\}$ can be formulated as:

$$\Lambda(\mathcal{R}) = \prod_{i=1}^{N} \frac{\dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(\dfrac{-(\mathcal{R}_i - m_i)^2}{2\sigma^2}\right)}{\dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(\dfrac{-\mathcal{R}_i^2}{2\sigma^2}\right)}. \tag{6}$$

Solving (6) using (3) we get:

$$\ln\eta \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\lessgtr}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left\{\mathcal{R}_i^2 - n^2\right\}, \tag{7}$$

where η is a tunable parameter.

In summary, n is the distribution of median-filtered baseline bandwidth values. As new bandwidth estimates arrive, they are plugged into the likelihood ratio defined in (7). The output of the test is then compared to an upper threshold η_1 and a lower threshold η_0. If $\Lambda(\mathcal{R}_i) \leq \eta_0$, we accept the null hypothesis $\mathcal{H}_0$. Alternatively, if $\Lambda(\mathcal{R}_i) \geq \eta_1$, we accept the alternate hypothesis $\mathcal{H}_1$. If neither case is true, we conclude that we do not have enough information to make a decision and wait for the next measurement to recalculate $\Lambda(\mathcal{R}_i)$.

C. Threshold Optimization

Wald showed [32] that we can define the thresholds η_0 and η_1 in terms of the rate of true-positive (or detection rate) $\mathcal{P}_D$ and the false positive rate $\mathcal{P}_F$. He showed that these rates may be approximated by user-defined values α and β such that:

$$\mathcal{P}_F \leq \alpha \qquad \text{and} \qquad \mathcal{P}_D \geq \beta. \tag{8}$$

[6] TCP Reno and its variations –which are loss-based approaches– are the most widely used TCP stacks.

Algorithm 2: Event detection.

```
Data: Array of performance measurements $\Omega$, False positive
      rate $\alpha$, Detection rate $\beta$, window size $\rho$, initial duration
      for training dataset $\delta$ and width of median filter $\nu$.
Result: Array of timestamp-brackets $\psi$ classifying windows as
        containing events.
Apply low-pass median filter of width $\nu$ to obtain $\Omega_{tr}$;
Compute $\mu_{tr}$ and $\sigma_{tr}$ for $\{\omega_t \in \Omega_{tr} | t_0 < t < t_0 + \delta\}$;
Let threshold $\eta_1 = \frac{\beta\sigma^2}{\alpha}$, $\eta_0 = \frac{1-\beta}{1-\alpha}$ and $t_0 = 0$;
                                  /* determine the baseline */
          /* initialize the observation window */
Let $\tau_s = t_0 + \delta - \nu$ and $\tau_e = t_0 + \delta$ ;
for $\{\omega \in \Omega\}$ do
    Let $x_1 = rand()$, $x_2 = rand()$ and
    $n = \sqrt{-2\ln(x_1)} \cdot sin(2\pi x_2) \cdot \sigma$;
    $R = median\{\omega_i | \tau_s \le i \le \tau_e\}$;
    Compute $\eta = \frac{R^2 - n^2}{2\sigma_{tr}^2}$ ;
    if $\eta_1 < \eta$ then
        Observation $\omega$ is anomalous, add $\omega's$ timestamp to the
        array of events $\psi$;
    else if $\eta < \eta_0$ then
        Observation $\omega$ is not anomalous;
        Update the training dataset with $\omega$, discard the oldest
        entry, recalculate $\sigma_{tr}$ and $\eta_1$;
    else Not enough information to make a decision;
    Increment $\tau_s$ and $\tau_e$;
end
Analyze $\psi$ and combine consecutive anomalous windows
defining unique events;
```

We set these values to $\alpha = 0.2$ and $\beta = 0.99$.

As an example, consider that the alternate hypothesis is accepted when it is in fact true; i.e. (7) met the threshold: $\eta_1 \le \frac{Pr(\mathcal{R}|\mathcal{H}_1)}{Pr(\mathcal{R}|\mathcal{H}_0)}$. This means that the detection rate $\mathcal{P}_D$ is at least η_1 times the false positive rate $\mathcal{P}_F$ when $\mathcal{H}_1$ is true. Consequently, we can define η_1 and η_0 as:

$$\eta_1 \le \frac{\mathcal{P}_D}{\mathcal{P}_F} \quad \text{and} \quad \frac{1-\mathcal{P}_D}{1-\mathcal{P}_F} \le \eta_0. \tag{9}$$

Using the approximation as defined in (8), we get:

$$\eta_1 = \frac{\beta}{\alpha} \quad \text{and} \quad \eta_0 = \frac{1-\beta}{1-\alpha}. \tag{10}$$

Using the above thresholds, we can derive the upper and lower thresholds corresponding to the two hypotheses from user-defined detection and false-positive constraints. However, from experiments we observe that the upper threshold of η_1 renders the algorithm too sensitive to datasets with large variance. In order to allow the algorithm to adapt itself to datasets in which the variance of the bandwidth measurements changes over time, we redefine the upper threshold η_1 as:

$$\eta_1 = \frac{\beta\sigma^2}{\alpha}, \tag{11}$$

where σ^2 is the variance of the median-filtered data being analyzed for anomalies. Step-wise execution of the above decision-theoretic approach to real-time path event detection is outlined in Algorithm 2.

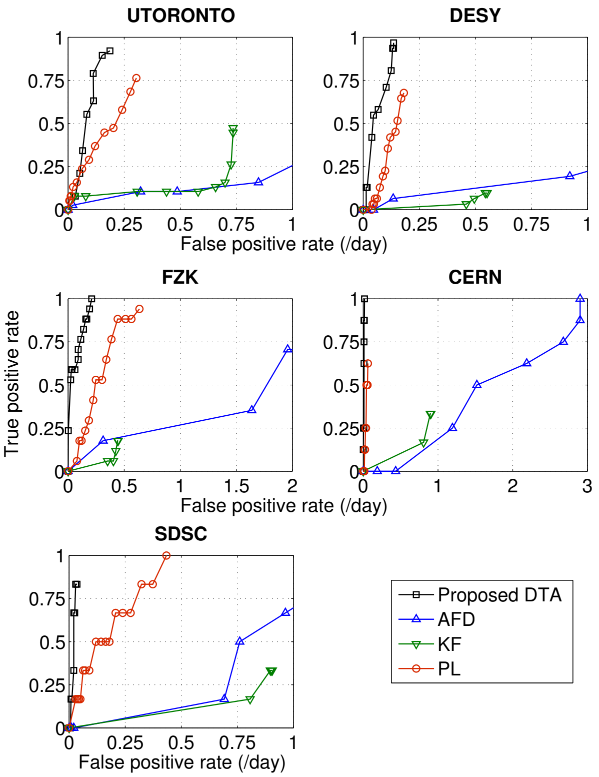


Fig. 6. Comparison of the Decision Theoretic Approach (DTA) with Kalman Filters (KF), Adaptive Fault Detection (AFD) and Plateau Algorithm (PL).

TABLE V
INPUT PARAMETERS (FALSE POSITIVE RATE, DETECTION RATE AND MEDIAN FILTER'S WIDTH) FOR THE DECISION THEORETIC APPROACH.

	α	β	n
UTOR	0.145	0.99	17
CERN	0.31	0.99	7
DESY	0.02	0.99	7
SDSC	0.29	0.99	15
FZK	0.05	0.99	15

D. Accuracy of the Proposed Approach

A comparison of the proposed Decision-Theoretic Approach (DTA) with the Adaptive Fault Detection (AFD), Kalman Filters (KF) and Plateau Algorithm (PL) is shown in Fig. 6. The input parameters listed in Table V were used by DTA to compile these results.

It is clear that the proposed approach provides consistently higher accuracy of change detection than all the existing methods. AFD and KF methods provide significantly lower detection accuracy than the proposed DTA because of the reasons enumerated in the last section. Plateau algorithm gives more false-positives than DTA because it operates on rigid thresholds given as input to the algorithm. Consequently, if an Internet path changes its characteristics even slightly (e.g., increases its variance,) Plateau is unable to adapt its parameters in accordance with the slightly modified baseline behavior. In summary, DTA and Plateau achieve high detection rates for acceptable levels of false-positive rates, whereas KF and

TABLE VI
AVERAGE DETECTION DELAY $\bar{\varepsilon}$ OF THE PROPOSED DTA APPROACH (IN TERMS OF ADDITIONAL OBSERVATIONS REQUIRED BEFORE AN EVENT IS DETECTED).

	Decision Theoretic Approach					
	Detected		*Undetected*		*Total*	*False positives*
	#	$\bar{\varepsilon}$	#	$\bar{\varepsilon}$		
UTOR	35	9.77	3	∞	38	168
CERN	8	7.25	0	∞	8	18
DESY	31	5.77	0	∞	31	91
SDSC	5	7.60	1	∞	6	14
FZK	17	22.13	0	∞	17	110

AFD methods achieve 100% detection rates at very high false-positive rates.

E. Detection Delay of the Proposed Approach

Detection delays of DTA are provided in Table VI. Comparison with Table III shows that DTA incurs nearly the same detection delays as that of PL and KF. DTA performs slightly better than PL on three links, but slightly worse on the other two. On the other hand, the KF method features lower delays, but it does so with poor true-positive rates. The AFD method presents the worst results in comparison to others. We therefore conclude that DTA does not present significant improvements in detection delays. Nevertheless, for detection delays comparable to existing approaches, DTA provides considerably higher accuracy.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we first evaluated the accuracies and detection delays of existing anomaly detectors for event detection over Internet paths. We concluded that existing methods can provide acceptable detection delays, but their accuracies are quite low. We then revealed statistical characteristics of Internet bandwidth measurements that can facilitate detection. Based on these characteristics, we proposed a decision-theoretic anomaly detector which could achieve consistently higher accuracy than the existing detectors while having similar detection delay as existing detectors. As an extension of this work, we are developing algorithms for event diagnosis and extraction of diurnal patterns.

ACKNOWLEDGMENTS

We acknowledge D. Loguinov, Y. Gu, L. Cottrell, Y. T. Li and G. Jared for their valuable comments on the subject.

REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *SIGCOMM*, 2005, pp. 217–228.

[2] T. Ahmed and M. Coates, "Multivariate online anomaly detection using kernel recursive least squares," in *INFOCOM*, 2007, pp. 625–633.

[3] P. Barford and J. Kline, "A signal analysis of network traffic anomalies," in *Internet Measurement Workshop*. ACM SIGCOMM, 2002.

[4] H. Ringberg, A. Soule, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *SIGMETRICS*, 2007, pp. 109–120.

[5] T. D. Lane, "Machine learning techniques for the computer security domain of anomaly detection," Ph.D. dissertation, Department of Electrical and Computer Engineering, Purdue University, Aug. 2000.

[6] L. Cottrell, "Internet end-to-end performance monitoring - bandwidth to the world (iepm-bw) project," 2002. [Online]. Available: https://confluence.slac.stanford.edu/display/IEPM

[7] W. Matthews and L. Cottrell, "The pinger project: active internet performance monitoring for the henp community," in *Communications Magazine*. IEEE, 2000, pp. 130–136.

[8] P. Calyam and A. Kalash, "Rice: A reliable and efficient remote instrumentation collaboration environment," in *IMMERSCOM*, 2007.

[9] C. H. Huang, "Biogrid: a collaborative environment for life science research," in *Anaesthesia, Pain, Intensive Care and Emergency A.P.I.C.E.* Springer Milan, 2006, pp. 123–132.

[10] D. Sisalem and A. Wolisz, "Towards tcp-friendly adaptive multimedia applications based on rtp," in *Proceedings of the The Fourth IEEE Symposium on Computers and Communications*. Washington, DC, USA: IEEE Computer Society, 1999, p. 166.

[11] M. Claypool and A. Tripathi, "Adaptive video streaming using content-aware media scaling." [Online]. Available: citeseer.ist.psu.edu/claypool04adaptive.html

[12] C. Logg and L. Cottrell, "Experiences in traceroute and available bandwidth change analysis," in *Proceedings of the ACM SIGCOMM workshop on Network Troubleshooting*. ACM, 2004, pp. 247–252.

[13] A. Soule, K. Salamatian, and N. Taft, "Combining filtering and statistical methods for anomaly detection," in *Internet Measurement Conference - IMC*. USENIX, 2005, pp. 331–344.

[14] H. Hajji, "Statistical analysis of network traffic for adaptive faults detection," in *IEEE Transactions on Neural Networks*, 2005.

[15] IPerf, "The tcp/udp bandwidth measurement tool." [Online]. Available: http://dast.nlanr.net/Projects/Iperf/

[16] pathChirp, "Bandwidth estimation tool." [Online]. Available: http://www.spin.rice.edu/Software/pathChirp/

[17] S. Shalunov, "thrulay: Network capacity and delay tester." [Online]. Available: http://shlang.com/thrulay/

[18] X. Liu, K. Ravindran, and D. Loguinov, "Multi-hop probing asymptotics in available bandwidth estimation: Stochastic analysis," in *Internet Measurement Conference - IMC*. USENIX/ACM, 2005.

[19] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-hop probing asymptotics in available bandwidth estimation: Sample-path analysis," in *Internet Measurement Conference - IMC*. ACM, 2004.

[20] S. Hares and C. Wittbrodt, "Essential tools for the osi internet," *Network Working Group - RFC 1574*, February 1994, informational.

[21] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas, "A one-way active measurement protocol (owamp)," *Network Working Group - RFC 4656*, September 2006, standards Track.

[22] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," in *ACM/IEEE Transactions on Networking*, August 2003, pp. 537–549.

[23] A. Lakhina, K. Papagiannaki, and C. Diot, "Structural analysis of network traffic flows," in *ACM SIGMETRICS*, 2004.

[24] A. J. McGregor and H.-W. Braun, "Automated event detection for active measurement systems," *Passive and Active Measurement Conference - PAM*, 2001, http://byerley.cs.waikato.ac.nz/ tonym/papers/event.pdf.

[25] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters, Science Direct*, pp. 861–874, 2006.

[26] L. Cottrell and C. Logg, "Evaluation of techniques to detect significant network performance problems using end-to-end active network measurements," in *Network Operations and Management Symposium - NOMS*, 2006, pp. 85–94.

[27] S. Floyd and T. Henderson, "The newreno modification to tcp's fast recovery algorithm," *Network Working Group - RFC 3782*, 2004.

[28] C. Jin, D. Wei, and S. Low, "Fast tcp: from theory to experiments," in *IEEE Network*, 2005, pp. 4–11.

[29] H. Bullot, L. Cottrell, and R. Hughes-Jones, "Evaluation of advanced tcp stacks on fast long-distance production networks," in *Journal of Grid Computing*. SpringerLink, November 2004, pp. 345–359.

[30] D. Heeger, "Signal detection theory," *Department of Psychology, New York University*, 2007. [Online]. Available: http://www.cns.nyu.edu/∼david/handouts/sdt/sdt.html

[31] H. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. John Wiley and Sons, 2001. [Online]. Available: http://gunston.gmu.edu/demt/demtp1/

[32] A. Wald, *Sequential Analysis*. John Wiley and Sons, 2004.