

Anomalous Event Detection for Internet End-to-end Performance Measurements

Abstract

We describe and compare the use of several different algorithms that we have implemented to detect persistent anomalous events in real end-to-end Internet performance measurements. The measurements are based on active probes running on 40 production network paths with bottlenecks varying from 0.5Mbits/s to 1000Mbit/s. For well behaved data (no missed measurements and no very large outliers) with small seasonal changes most algorithms identify similar events. We compare the algorithms' robustness with respect to false positives and missed events especially when there are large seasonal effects in the data. We also discuss the applicability of the algorithms in terms of their intuitiveness, their speed of execution as implemented, and areas of applicability.

Keywords

Anomalous event detection, forecasting, network monitoring, network performance, performance analysis, persistent anomalies, trouble shooting, Kolmogorov-Smirnov, Holt-Winters, Principal Component Analysis, plateau algorithm.

Authors

R. Les. Cottrell, Connie Logg and Mahesh Chhaparia are with SLAC, 2575 Sand Hill Road, Menlo Park, CA 94025, USA,

Ph #: +1-650-926-2523, Fax #: 1+650-926-2523

Email: {cottrell,cal,maheshkc}@slac.stanford.edu

Maxim Grigoriev is with the Fermilab, Batavia, IL 60510, USA

Fermilab, MS 368, PO BOX 500, Batavia, IL 60510-0500

Ph #: +1-630-840-6024

Email: maxim@fnal.gov

Felipe Haro is with Pontificia Universidad Catolica de Chile, Vicuna Mackenna, Santiago, Chile

Onofre Jarpa 10107-G, Santiago, Chile

Ph #: +(56-2) 2737163

Email: felipeharo@gmail.com

Fawad Nazir is with the NUST Institute of Information Technology

NIIT 166-A, Street # 9, Chaklala Scheme # 3, Rawalpindi, Pakistan

Ph #: +92-51-9280658 , Fax #: +92-51-9280782

Email: fawad.nazir@niit.edu.pk

Mark Sandford is with the Department of Electronic Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK

Ph #: +44(0)1509 227079

Email: J.M.Sandford@lboro.ac.uk

countries, the static bottlenecks vary from 0.5Mbits/s to 1000Mbits/s, the paths traverse about 50 Autonomous Systems (ASs) and over 15 major Internet Service Providers (ISPs). The topology of the remote hosts is seen in Fig. 1. The main ISPs that the paths cross are identified as shaded boxes. For Abilene and ESnet the major Points of Presence (PoPs) are also identified. The remote host sites are also noted, as well as the capacity bottlenecks (Cap) for the paths. Five of the remote hosts (identified in Fig. 1 by “I2” and “Host”) are at ISP PoPs, the remainder are at end user sites.

The measurements also suffer from gaps in the observations due to problems with the measurement host, the paths and/or the remote (measured host).

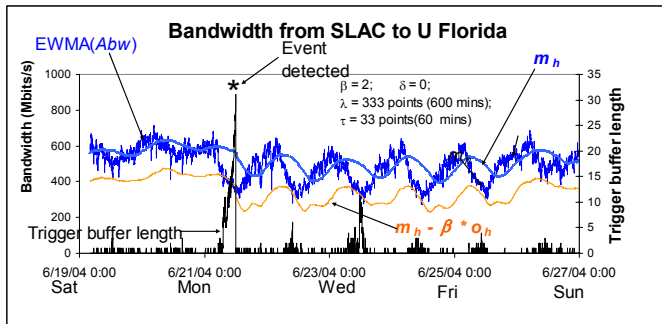


Figure 2 : ABwE bandwidth estimates from SLAC to U. Florida with a history buffer duration (λ) of 10 hours.

III. ANALYSIS

A. Plateau Algorithm

The “Plateau” bandwidth change detection algorithm is described in [14]. It is a modification of the algorithm described in [15] that was used to detect step changes in a time series set of measurements of Round Trip Times (RTT). Here we use it to analyze both the Abw and Cap measurements. Currently, missing measurements (e.g. because there is no functioning path between the monitor and monitoring host) are ignored compressing time so the gap is covered over.

The plateau algorithm basically divides the measurements into two buffers: a history buffer (h) for base-lining, or into a trigger buffer (t), when a measurement meets a specific requirement. The specific requirement is that the current measurement is less than β (we use bold face to indicate user settable parameters) standard deviations (σ_h) below the current mean of the history buffer m_h . If the measurement is placed in h then the oldest entry is removed from t . The buffers have maximum lengths of λ (history) and τ (trigger). Given a requested buffer duration, the number of items in a buffer (length) is calculated using the median time separation of the data points. When τ is reached the mean of the trigger buffer m_t is compared with m_h and if the relative difference $\Delta = (m_h - m_t) / m_h$ is greater than the threshold δ then an event is deemed to have occurred.

Sensible values of β are between 2 and 3 [16], we used $\beta = 2$. To minimize the effects of diurnal changes we used $\lambda = 1$

day. Longer values flatten the time series behavior of m_h , shorter values result in the sinusoidal-wave like curve of m_h being out of synchronization with the diurnal changes (see for example Fig. 2 (where $\lambda = 10$ hours, $\tau = 3$ hours, $\delta = 0\%$), and is seen to trail the EWMA(Abw) by several hours). Note that missing data points can also cause loss of synchronization. Since we were only interested in long term changes we typically use $\tau = 3$ hours. We currently use $\delta = 33\%$. Larger values of δ are likely to miss more real events; lower values are likely to lead to more false positives.

B. Kolmogorov-Smirnov (KS)

The KS test [17] is the best known of several distribution free techniques for testing for general differences between distributions. The technique makes no assumption about the underlying distribution of the measurements. It compares the observed and expected Cumulative Distribution Functions (CDF) for M data points before (expected) and after (observed) each measurement being evaluated. The KS test is calculated taking the vertical difference between the two CDFs as a test statistic. M in the current work was chosen to be 100 (5 hours for measurements separated by 3 minutes) as a reasonable compromise between the accuracy obtained (sufficient points for the distributions) and the time to wait for a response or the analysis time. If response time is deemed less important it may well be worth using a larger value of M as this will provide a larger sample for each distribution. For previous work [18] with less frequent measurements, we chose M to be 24 hours worth of data to minimize the diurnal effects as discussed earlier in the current paper (although one still gets false positives associated with weekends, public holidays etc.)

C. Holt-Winters (HW) Algorithm

The Holt-Winters (HW) [19] [20] algorithm uses a triple EWMA approximation to characterize the time series behavior as a superposition of three components: a baseline, a linear trend and a seasonal effect (e.g. diurnal changes). We developed two implementations of the HW technique and also used the RRD implementation [19] to compare our results against, and to understand the technique. We will focus our discussion on the implementation developed at SLAC (based on the formulation in [20]) since it has the most flexibility for our needs.

HW is critically dependent on having regularly spaced data with no missing points, so the first step is to bin the data into regularly spaced time bins and use similar data to interpolate for bins with no data. For bins with no data in the first week we use data from following weeks for the same day and time bin. For the following weeks we use the previous week’s interpolated data. For our data with bin widths of 3 minutes we found that having about five to seven weeks of data enabled us to successfully interpolate the data and fill in missing bins. Once we have the first week’s interpolated data, new data can be quickly merged onto the existing interpolated data without having to go back through all the data.

Due to the noisiness of the data, we also set the maximum

forecast = maximum of all observed values.

We used two methods to choose the initial HW parameters.

- We chose the initial HW parameters using the guidelines in [18]. 99% of the contribution for the baseline EWMA came from measurements made in the last 24 hours; 99% of the seasonal EWMA contribution came from the last week, 50% of the trend EWMA contribution came from last 24 hours.
- For each path, we minimized the sum of the squares of the residuals ($R^2 = \sum r_i^2$ where $r_i = y_i - f_i$, the sum is over all interpolated data, y_i is the interpolated observation at time i and f_i is the forecast at the same time) as a function of the HW parameters. This method always resulted in the trend parameter being set to very close to 0 (< 0.00001).

We have settled on using the second method. It provides good forecasts, works for a wider range of paths and requires minimal user input.

Several methods were tested to raise anomalous event conditions:

- The residual (r_i) at each point was examined to see if it was a trigger, i.e. outside the standard deviation of the forecast for the last 100 points. If 70% of the points in a window of 2.5 hours were triggers then an event was generated.
- The residual was compared to the EWMA of the absolute deviation [18]. If over 82% of the residuals were outside twice the EWMA of the absolute deviation in the last 84 minutes then an event was generated.
- With a moving window sized to cover 12 hours we calculated $X^2 = \sum r_i^2 / f_i$, and using tabulated X^2 values for $N-1$ degrees of freedom, where N is number of points per time window we set a threshold to generate our triggers and if there were more than 50% of points in the time window generated triggers then alert was raised.
- We applied the plateau algorithm and KS to the HW residuals.

D. Mark Burgess (MB) Technique

The Mark Burgess (MB) technique introduces a two dimensional time approach [21] to classify a periodic, adaptive threshold for service level anomaly detection. An iterative algorithm is applied to history analysis on this periodic time to provide a smooth roll-off in the significance of the data with time. This method was originally designed to detect anomalous behavior on a single host, with the aim of using the information for self-regulation, by initiating a counter response. An anomaly is indicated by a code indicating the state of the given statistic, as compared to an average of equivalent earlier times.

IV. RESULTS

To provide a canonical set of measurements to evaluate and compare the detection methods against we used *Cap* measurements from SLAC to 30 remote hosts for ~100 days from June through September 2004.

To first characterize the events seen in the canonical data, we used the Plateau algorithm, since it is the most intuitive of the algorithms and allows direct variation of parameters representing the size and duration of an event. We set δ to 0 (i.e. we detect all events that fill the trigger buffer) and the other user parameters set as described above. About 25 of the 40 paths manifested one or more events in this period. We carefully reviewed each of these events and created a library of interesting events. We observe three general types of events that trigger our Plateau algorithm.

- Step down changes in bandwidth (“step”)
- Diurnal changes (“diurnal”)
- Changes caused by actions causing congestion, e.g. a regularly scheduled cron job (“host”), or network bandwidth test, flash crowds etc.

Three paths out of 30 (Caltech, NIIT and U. Florida) exhibited marked diurnal changes that triggered “diurnal” plateau events, especially following a weekend. To study the diurnal behavior more carefully we binned the bandwidth data by hour of day and calculated the percentiles to identify the daily bandwidth patterns. An example is shown in Fig. 3 where there is a quick decrease in bandwidth when people arrive to work (20:00-23:00 PDT = 08:00 – 11:00 Pakistan time). This in turn causes an abnormally high number of events to be detected by the plateau and KS algorithms during these hours (see Fig. 4). Usually these changes are not as sudden as the typical step change which may also help in separating the two types. For our purposes, these “diurnal” events are false positives that need to be eliminated.

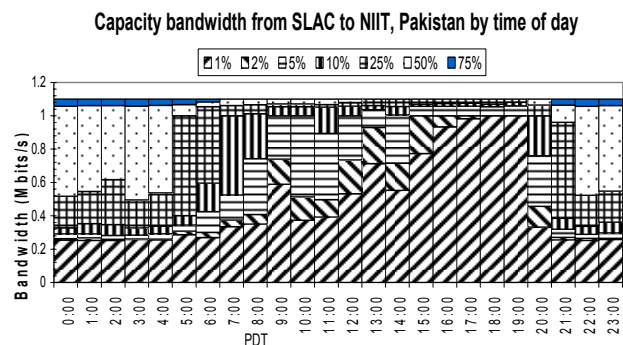


Figure 3: Percentiles of capacity bandwidth seen on the SLAC-NIIT path as a function of time of day

One host (ANL) exhibited regular “host” type events that were tracked down to a cron job running on the host that used (via NFS) the network heavily. This host was eliminated from further non-seasonal analyses. Events for a given host typically have a small range for Δ (standard deviation (Δ) / mean (Δ) $\sim 0.11 \pm 0.1$) indicating that the backup routes or diurnal behavior is consistent. This manifests itself in a multimodal Distribution Function for Δ .

By careful examination of ~ 120 plateau candidate events detected² with $\delta = 0$ (and ignoring whether the events are diurnal) we classify all candidates as to whether they are

² Less than 10% of these candidate events were associated with noticeable traceroute changes.

events we are interested³ in or not (i.e. exhibit sharp drop in bandwidth (e.g. 90% of change occurs in < 220 mins), persist for a long term ($\gg 3$ hours) and are large enough (e.g. $\delta > 10\%$). With $\delta = 10\%$ and restricting the duration of 90% of the trigger buffer to 220 minutes, we miss 8% of the events and see 16% false positives. Increasing δ to 33% we get 32% misses and 2% false positives. In this case 15% of the events are caused by diurnal changes.

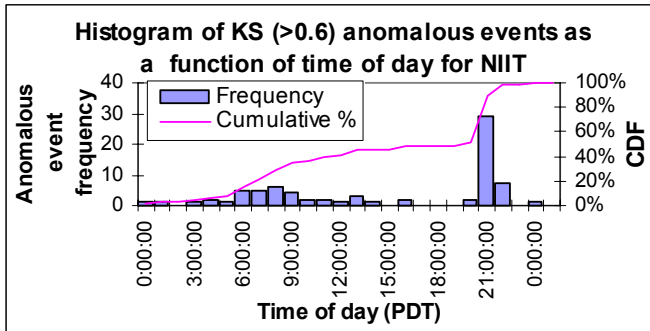


Figure 4: Distribution of events as a function of time of day for the SLAC-NIIT path.

We then also applied KS and HW algorithms to the canonical data. Fig. 5 shows a visualization of the plateau, KS and HW algorithms applied to capacity bandwidth observations with step changes in performance from SLAC to BINP. As expected all three algorithms detect marked changes. KS detects both increases and decreases in the data and thus detects roughly twice as many events (229:116) as the plateau algorithm which is currently tuned for negative changes only. Detecting both steps is valuable since, for example, it enables determining the duration of a change and/or taking some action when the original performance is restored. KS also provides the most accurate time estimate⁴ for when the event occurred which is important when correlating the event with other time-dependent information.

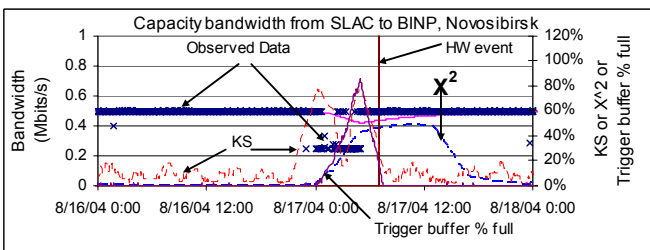


Figure 5: Plateau, KS and HW algorithms applied to an observed step down and back up performance change

In Fig. 5 the KS statistic (dotted) is seen to reach a value of about 0.72, on both the step up and down. The plateau algorithm's trigger buffer (solid line) only reaches $\sim 80\%$ full (since the step down's duration is too short, in this case ~ 4 hours) so no event is triggered. The HW X^2 (dashed line)

³ Others may have very different criteria, in particular the duration of the change or the magnitude of the drop. With the exception of the plateau algorithm, the duration and magnitude filters need to be applied as a separate step.

⁴ The time estimates for plateau and HW could be improved, for example for the plateau algorithm by identifying the time of the change when the event occurred as the time at which the trigger buffer reached say 10% full.

triggers an event after the performance has recovered.

Increasing the threshold value (K) of the KS coefficient that defines an event reduces the false positives at the cost of increasing the missed events as seen in Fig. 6. Currently we are using a value of $K = 0.7$ to trigger an event. Most (69%) of the false positives come from four paths. Minimizing R^2 to estimate the initial HW parameters results in a fairly wide range of values of the local smoothing parameter ($a = 0.0001$ to 0.95, median = 0.0024 ± 0.12) and the seasonal parameter ($b = 0.0023$ to 0.999, median = 0.22 ± 0.2). About 50% of the paths have $0.0008 < a < 0.02$ and $0.18 < b < 0.4$. There is very weak correlation between a and b or R^2 and a or b , which is suggestive that there may not be suitable single set of parameters for all paths.

Using HW to forecast, and with the 70% of triggers in 2.5 hours method to detect events, plus bunching together events separated by < 3 hours, successfully removed the diurnal events for Caltech, NIIT and U. Florida. It also succeeded in eliminating the effects of the ANL cron jobs that ran at regular times each morning. However, there appear to be similar host based effects that do not occur at regular intervals that make the ANL data problematic. A similar effect giving rise to false positives was seen with the SDSC path. In this case there was a step down of about 10% (65Mbits/s) lasting for 3-4 hours starting around 1am each day for about one week.

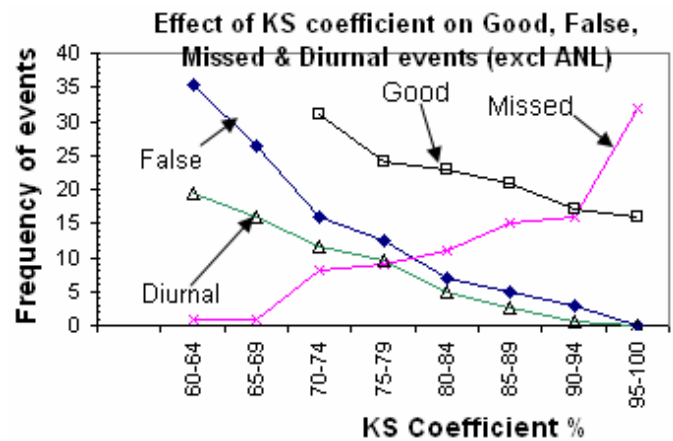


Figure 6: Cumulative step down event types as function of KS coefficient

With the above HW event detection method, three paths (SOX, NASA and CESnet) with very small deviations in the observations, had small changes in bandwidth that resulted in events with very small changes in bandwidth.

Eliminating the ANL and SDSC paths, and demanding a bandwidth change of at least 5% for an event, HW detected 23 true events, with 1 false positive and 6 missed events. Four of the misses were during the first week of data, when the HW algorithm performs poorly due to not having good initial estimates. A further miss was since the change happened slowly (it took over a day to get from the initial value to the new stepped down value). The final miss was for a step down in performance that only lasted four hours. Thus, with the caveat that HW cannot forecast (and thus not amenable to event detection methods discussed above) for dataset affected

by applications running at irregular times and causing congestion, the HW technique works well for our data. To understand such events better more measurements are needed on hosts and network equipment to isolate the cause.

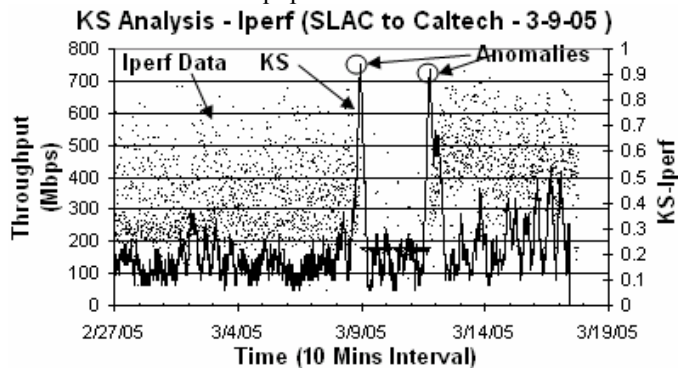


Figure 7 : KS on Iperf data from SLAC to Caltech

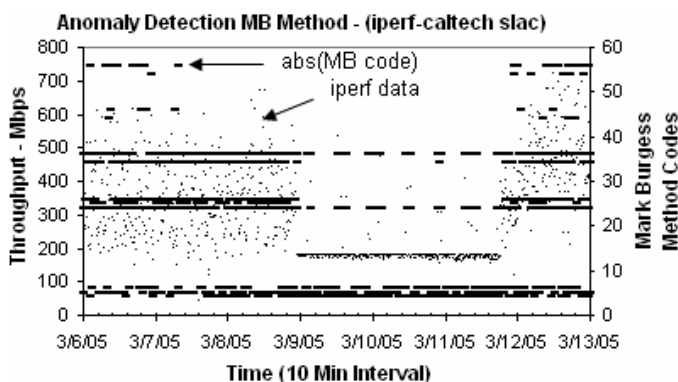


Figure 8 : MB on Iperf Data SLAC to Caltech

Figs. 7 and 8 show the effects of applying the KS and MB techniques to the same iperf data (from SLAC to Caltech March 6 – 13, 2005). Fig. 8 is a snapshot of the original graph showing the MB technique applied on Iperf Data. This snapshot gives a good idea about the real-time behavior of MB technique. It is seen in Fig. 7 that the KS coefficient (solid line) identifies the obvious long term step down and step up (both identified by circles) in the data on March 9, 2005 and March 11, 2005 respectively. In Fig. 8 on the other hand, the MB code (light dots) oscillates wildly as it tries to track the individual spikes in the noisy data and misses the important steps down and up.

We also applied plateau and KS on the HW residuals for the U. of Michigan dataset which has both diurnal changes and one-off anomalous events. We find that although both are able to detect one-off step-downs, plateau shows no false positives while KS raises several events on weekends in diurnal region. This happens because KS compares two frequency distributions irrespective of the relative change in values. During weekends, residuals are close to 0 as data values usually mirror past weekend's data and HW is able to make good forecasts based on its past week's seasonal cycle values. However due to higher network usage during weekdays, there are higher fluctuations and residuals, though small in absolute value, are more spread out. So KS on weekend data effectively compares two very different distributions (past weekdays and weekend) thereby raising

false events. This interesting observation highlights a weakness of KS for our current application.

V. CONCLUSIONS

For measurements with limited diurnal (or other seasonal) changes the Plateau, KS and HW algorithms work well.

The plateau algorithm is easily understood by people with a non-statistical background and has easy to interpret user settable parameters.

The KS algorithm is the most statistically formal of the three algorithms. It has only two parameters: the number of data points (M) used to evaluate the distribution functions; and the threshold (K) of the KS coefficient above which an event is assumed to have occurred. It provides the best estimates of when a step occurs.

The MB technique is mainly aimed at providing real time anomaly detection and thus is not suited to finding long-term persistent anomalous changes.

For the HW techniques it is critical to provide complete data at regular time intervals. Also one needs a few weeks worth of data to get a reasonable estimate of the seasonal variations. Estimating the parameters using the minimization techniques appears to work well in most cases. The basic Holt-Winters algorithm is a forecasting technique, and so may also be used for providing forecast information to applications such as Grid middleware [23].

The HW technique explicitly incorporates seasonal changes and so event detection methods with HW work better than the plateau and KS methods on paths with significant diurnal changes. Among the different event detection methods discussed for HW, plateau on HW residuals is the most effective in identifying seasonal variations unlike KS on HW residuals which generates several events due to its inherent weakness in simply comparing cumulative distributions irrespective of relative change in data values. Each technique requires several parameters. The HW parameters require no user input and hence may be applied with relative ease to normalize the data with respect to seasonal changes.

Both the plateau and HW algorithms are implemented on Linux systems as Perl scripts and so should be relatively easy to port. Currently no attempt has been made to optimize the speed of execution. For 43K data points on a dual Xeon 3GHz cpu host, it takes about 15 seconds to interpolate the data, 30-40 seconds to minimize R^2 to find the optimum HW parameters, and about 2 minutes for the HW analysis and reporting. The plateau algorithm has mainly been used for exploring the data and thus has many extra tests and reporting which result in it taking about six times as long. The KS algorithm is implemented in C and takes about 1 minute with $M = 100$. The execution time of KS, however, is very dependent on M . For example, increasing M from 100 to 400 increases the analysis time by a factor of 14.

We are looking at using wavelet decomposition to eliminate outliers and white noise seen in real data.

The subspace PCA analysis has been reported [24] to work well when applied to measurements from core routers. It is

unclear how well it will work on less correlated end-to-end active Internet performance measurements. It has the advantage of being able to simultaneously look at measurements of multiple metrics (e.g. RTT, iperf throughput, *Cap*, *Xtr*, reverse and forward performance measures such as provided by ABwE, and/or hosts/system performance measures) and paths simultaneously. On the other hand it is less intelligible to someone without a statistical background. We have implemented this algorithm and are currently applying it to the ABwE data.

A second, filter process may be applied once potential events have been identified using one of the above techniques. A filter may serve two purposes. Firstly unwanted events, or false positives may be removed and not reported. Secondly, events that are noteworthy may be classified to aid in further diagnostic processes. A neural network has been used as such a filter process in [18]. We are also looking at using neural networks to interpolate between infrequent, heavyweight measurements such as iperf by using more frequent lightweight measurements such as ABwE or RTT etc.

Once we have a robust, reliable anomalous event detection technique that works with the ABwE data, we will extend it to other measurements including GridFTP and observations of host/system performance measures. We also plan to use the current work to generate alerts. These will be filtered relevant information gathered from network devices, analyzed and reported to network administrators.

Besides detecting anomalous events, we plan to make the forecasts available for Grid middleware such as a replication manager.

ACKNOWLEDGMENT

We gratefully acknowledge Jerrod Williams and the administrators at the remote hosts for their work in setting up the hosts and keeping the measurements running. We also thank Mark Crovella, Jerry Friedman, and Waqar Mahmood for useful discussions on the subspace PCA technique, and Alf Wachsmann for help with the RRD version of HW and assistance with the CFETool version of MB.

REFERENCES

- [1] McGregor A, Braun H-W, and Brown J, "The NLANR network analysis infrastructure", *IEEE Communications Magazine*, May 2000
- [2] Cottrell R. L., Logg C, and Mei I-H, "Experiences and Results from a New High Performance Network and Application Monitoring Toolkit", *PAM 2003*.
- [3] Matthews W. and R. L. Cottrell, "The PingER Project: Active Internet Performance Monitoring for the HENP Community", *IEEE Communications Magazine*, May 2000.
- [4] "National Internet Measurement Infrastructure", available at <http://www.ncne.nlanr.net/nimi/>
- [5] "E2E piPES", available at <http://e2epi.internet2.edu/pipes/>
- [6] "RIPE NCC Test Traffic Measurements", available at <http://www.ripe.net/ttm/>
- [7] "MonALISA: Monitoring Agents using a Large Integrated Services Architecture" available at <http://monalisa.caltech.edu/>
- [8] Navratil J and Cottrell R. L., "ABwE: A practical Approach to Available Bandwidth Estimation", *PAM 2003* also SLAC-PUB-9622.
- [9] Alok Shriram , Margaret Murray, Young Hyun, Nevil Brownlee, Andre Broido, Marina Fomenkov, k claffy "Comparison of Public End-to-end Bandwidth Estimation Tools on High-Speed Links", *PAM2005*.
- [10] "Iperf the TCP/UDP Bandwidth Measurement Tool", available <http://dast.nlanr.net/Projects/iperf/>
- [11] "BBFTP Large Files Transfer Protocols", available at <http://doc.in2p3.fr/bbftp/>
- [12] "Globus Grid FTP Protocol and Software", available at <http://www.globus.org/datagrid/gridftp.html>
- [13] Grigoriev M, Cottrell R. L. and Logg C., "Wide Area Network Monitoring System for HEP Experiments at FNAL", *Computing in High Energy Physics*, Interlaken Switzerland, Sep 2004.
- [14] Logg C, Cottrell R. L. and Navratil J., "Experiences in Traceroute and Available Bandwidth Change Analysis", *SIGCOMM'04 Workshops*, Aug 30 & Sep 3, 2004, Portland OR, USA
- [15] McGregor A.J. and Braun H-W, "Automated Event Detection for Active Measurement Systems", Proceedings of PAM2001, Amsterdam, Netherlands, April 2001
- [16] A. Ward, P. Glynn and K. Richardson, "Internet service performance failure detection" *Performance Evaluation Review*, 26:38-43, 1998
- [17] Brockwell P. and Davis R, "Introduction to Time Series and Forecasting", *Springer* New York, 1996
- [18] Sandford, J.M., Parish, D.J. and Phillips, I.W., "Neural approach to detecting communication network events", *IEE Proc. Communications*, 1495 , October 2002, pp 257-264, ISSN 1350-2425.
- [19] Brutlag J.D., "Aberrant Behaviour Detection in Time Series for Network Monitoring", Proceedings of LISA 2000, New Orleans, LA, USA, December 2000.
- [20] NIST e-handbook of statistics, <http://www.itl.nist.gov/div898/handbook>.
- [21] Mark Burgess, Two dimensional time-series for anomaly detection and regulation in adaptive systems, Proceedings of 13th IFIP/IEEE International Workshop on Distributed System, operations and management (DSOM 2002). "Management Technologies for E-Commerce and E-Business Applications" Springer 2002.
- [22] Comparison between Two Dimensional time-series & KS technique for Anomalous Variations in End-to-End Internet Traffic : <http://www-iepm.slac.stanford.edu/monitoring/forecast/ksvsmb/ksvsmb.htm>
- [23] Ian Foster, Carl Kesselman, "The Grid 2: Blueprint for a New Computing Infrastructure", Publisher: Morgan Kaufmann; 2 edition (November 18, 2003), ISBN: 1558609334
- [24] Lakhina A, Crovella M, Diot C, "Diagnosing Network-Wide Traffic Anomalies", *Sigcomm* 2004.