Comparison of Approaches to Quantify Alert Conditions in Internet End-to-end Performance Measurements

R. Les Cottrell (SLAC), Maxim Grigoriev (Fermilab) and Connie Logg (SLAC)

***Abstract*—We describe and compare the use of two different algorithms to detect persistent anomalous events in end-to-end Internet performance measurements. The measurements are based on active probes running from two production network monitoring sites and the algorithms are embedded into the measurement infrastructure. The measurements include multiple metrics and are made at widely different intervals (1-3 minutes and 90-120 minutes).**

***Index Terms*—anomalous event detection, network monitoring, network performance, performance analysis, persistent anomalies, trouble shooting.**

I. Introduction

Management of wide area networking from an end user/administrator point of view is increasingly hard as the complexity of the paths, the diversity of the performance, and the dependency on the network increase. Several monitoring infrastructures have been built [1], [2], [3], [4], [5], [6], [7] to assist by addressing the measurement, archiving, analysis, and presentation aspects of end-to-end performance monitoring. Each of these infrastructures consists of tens to hundreds of monitoring hosts. Each of these monitoring hosts, can make measurements of multiple metrics e.g. delays (both Round Trip Time (RTT) and one way delay), loss, jitter, TCP achievable throughput, available bandwidth, and applications performance (e.g. file transfers or web requests) to hundreds of monitored (remote) hosts. Typically for every pair of hosts (monitor and remote host) there will be a time series plot for each metric, amounting to hundreds to thousands of plots that need to be reviewed to look for anomalous changes in performance. The network administrator can, at best, review some of these reports reactively upon being presented with a problem by a user. What is needed is to enable the network administrator to be pro-active and spot the problem before the user. This in turn needs a way to automatically and reliably (few false positives and most events detected) detect persistent, anomalous (unusual and significant) changes (events) in performance and report them in an efficient way to the network administrator.

In this paper we currently report on two approaches (by the time it is published we expect to report on four) to the problem of automatically detecting persistent anomalies in two different end-to-end network performance metrics using active end-to-end network performance measurements from two instantiations of the IEPM-BW [2] measurement infrastructure at two sites. The requirements for both cases are to detect decreases in performance that are sufficiently large and persist for sufficient time to be able to review the change and report the problem to the up stream provider's Network Operations Center.

The rest of the paper is organized as follows. Section II describes how the measurements were made, section III describes the analysis used to extract anomalous events, section IV describes the results, section V describes work in progress that will be reported in the final paper, and section VI presents the conclusions.

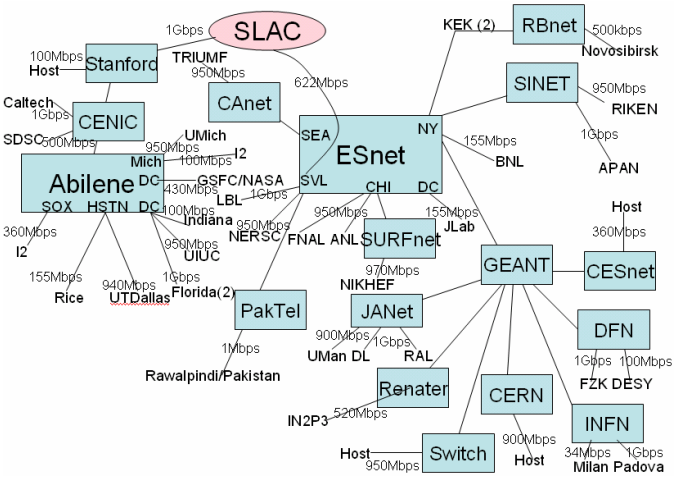

Figure 1: Topology of the remote hosts measured from SLAC.

II. Measurements

We use measurements from the ABwE [8] lightweight bandwidth estimation tool that uses the packet pair dispersion

Manuscript received October 8, 2004. This work was supported in part by the Director, Office of Science, Office of Advanced Scientific Computing Research, Mathematical and Computational Sciences Division under the U.S. Department of Energy. The Stanford Linear Accelerator Center (SLAC) is operated by Stanford University for the U.S. Department of Energy under contract DE-AC02-76SF00515. The Fermi National Accelerator Laboratory (Fermilab) is operated by Universities Research Association, Inc. for the U.S. Department of Energy under contract DE-AC02-76CH03000.

R. Les. Cottrell and Connie Logg are with SLAC, 2575 Sand Hill Road, Menlo Park, CA 94025, USA (+1-650-926-2523, fax 1+650-926-2523, email: {cottrell, cal}@slac.stanford.edu)

Maxim Grigoriev is with the Fermi lab, Batavia, IL 60510, USA (email: maxim@fnal.gov).

technique. Twenty packet pairs are used per direction for each measurement. The frequency of the measurements is one to three minute intervals. For each interval, three metrics are measured: dynamic bottleneck capacity (*Cap*) by analyzing the minimum packet pair separation; Cross Traffic (*Xtr*) by analyzing the packet pair dispersion; and the Available Bandwidth (*Abw*) = *Cap* – *Xtr*.

Measurements are also used from 10-15 second multi-stream iperf [9] tests, ping probes, BBFTP [10] and GridFTP [11] real file transfers every 90 minutes from SLAC and every 2 hours from Fermilab to each monitored host [12] for the monitoring site. All tests are run in both directions. The iperf measurements are considered the most accurate indicator of the TCP achievable throughput.

Upon completion of the analysis the list of alerts is combined and notification messages are sent to Network admins (Fermilab) or developers (SLAC).

The measurements made from SLAC are to about 40 hosts in 13 countries and 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 Fermilab measurements are to about 14 hosts in 5 countries, 6 of the remote sites are different from the SLAC sites. See Fig. 2 for the topology of remote hosts, monitored from Fermilab.

III. Analysis

A. Plateau Algorithm

The bandwidth change detection algorithm is described in [13]. It is a modification of the "plateau" algorithm [14] to detect step changes in a time series set of measurements. We analyze both the *Abw* and *Cap* measurements. The *Abw* measurements are probably of most interest to a user, however they are more sensitive to cross-traffic over which we have little control. Changes in *Cap* on the other hand are more likely to reflect route changes or operator errors etc. and thus may be easier to address. *Cap* estimates are thus generally preferred for our work. Since only 20 packet pairs are used for each bandwidth estimate, the statistical variability of the estimates is quite high. Estimates can thus vary dramatically from minute to minute and have large outliers. Therefore, ABwE also provides smoothed data using an Exponential Weighted Moving Average (EWMA) [15]. 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.

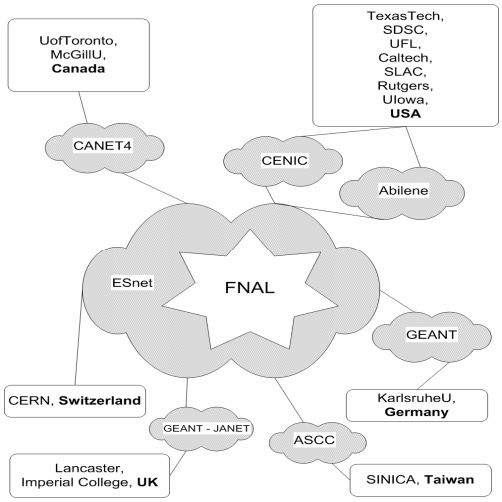

Figure 2: Sites monitored from Fermilab

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** standard deviations ($\mathbf{o_h}$) below the current mean of the history buffer $\mathbf{m_h}$. 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 $\mathbf{m_t}$ is compared with $\mathbf{m_h}$ and if the relative difference $\mathbf{\Delta = (m_h - m_t) / m_h}$ is greater than the threshold **δ** then an event is deemed to have occurred.

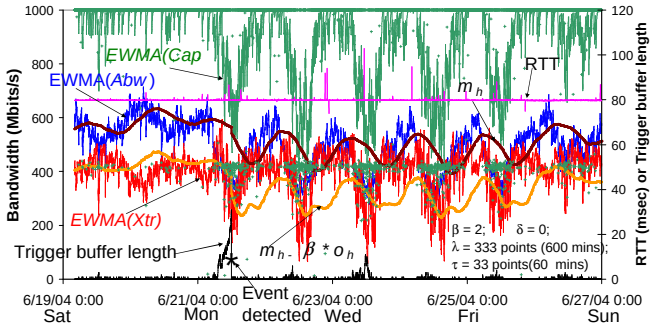

Figure 3: ABwE bandwidth estimates from SLAC to U Florida with a history buffer duration(λ) of 10 hours.

We have experimented with the user settable parameters. We used the default setting of **β** = 2. To minimize the effects of diurnal changes we used **λ** = 1 day. Longer values flatten the time series behavior of $\mathbf{m_h}$, shorter values result in the sine-wave like curve of $\mathbf{m_h}$ being out of synchronization with the diurnal changes (see for example Fig. 2 where $\mathbf{m_h}$ = 10 hours and is seen to trail the EWMA(*Abw*) by several hours). Since we were only interested in long term changes we used **τ** = 3 hours. We currently use **δ** = 33%. Larger values of **δ** are likely to miss more real events, lower values are likely to lead to more false positives.

B. Holt-Winters (HW) Algorithm

After researching publications on statistical network

analysis and forecasting techniques [14], [16], [18], the tri-exponential approximation (additive HW forecasting) with a moving time frame of the measurements with special rules was developed. It is based on combining the forecasting techniqu e, employing the triple-exponential smoothing as described by [16], [17] with the $\mathbf{X^2}$ error estimation method, and currently applied to iperf measurements normalized to the range 0-100. See Fig. 4 for a visualization of the bandwidth analysis, from Fermilab to McGill University, Canada. The triangles indicated detected "drops" in performance. The asterisks indicate lost measurements. Every forecasted value is a superposition of the seasona l trend (e.g. diurnal changes), trend over time (for example the increase in throughput demand) and the baseline. All parameters in the forecasting equations are chosen to give more weight to the most recent measurements. The initial set of par ameters is chosen according to [16], to adequately present mild trend and average seasonal variations. In the case of missing (lost) measurements the same forecasting technique is used to fill in gaps. It may be n oted, that additional calculations would be required to choose the most correct forecasting parameters set.

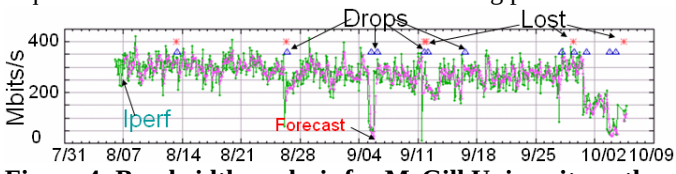

Figure 4: Bandwidth analysis for McGill University path from Fermilab

The $\mathbf{X^2}$ criterion is applied to every measurement from the moving window of the last N measurements (2xN hours with the current iperf measurement frequency). To raise an anomalous event condition, a set of special rules and procedures was developed. First of all, the system checks for the lost measurements and reports them, then for every new measurement the set of forecasted values for $N_{total} - N_{time\text{-}window}$ is built and for each measurement from $N_{time\text{-}window}$ the $\mathbf{X^2}$ estimate is calculated.

The zero de viation of the current measurement from the forecasted estimate is chosen as the null hypothesis . An anomalous event is generated if the $\mathbf{X^2}$ sum for the whole window ($N_{time\text{-}window} - 1$ degrees of freedom) lies outside of a 5% confidence interval. If there are $N_{time\text{-}window} - 1$ consecutive anomalous events (exact number defines the total sensitivity), then the system generates an alert and sends a message to the sysadmin, notifying about a significant and consistent drop in the network performance

IV. Results

A. Plateau Algorithm

We analyzed *Cap* measurements from SLAC to all 40 remote hosts for ~ 100 days from June through September 2004. With $\boldsymbol{\delta}$ set to 0 (i.e. we detect all events that fill the trigger buffer) and the other user parameters set as described above, about 50% of the hosts manifested one or more events in this period. We carefully reviewed each of these events and created a librar y of interesting events. We observe three general types of events triggered by our plateau algorithm.

- Step down changes in bandwidth ("step")
- Diurnal changes ("diurnal")
- Changes caused by known events causing congestion, e.g. a regularly scheduled cron j ob, or network bandwidth test ("host")

Three hosts out of 40 exhibited 11 marked diurnal changes that triggered "diurnal" events. These are false positives that need to be eliminated. Sixteen hosts with less marked diurnal changes exhibited 16 "step" change events. One host (ANL) exhibited regular "host" type events that were tracked down to a cron job running on the host soon after midnight each night that used (via NFS) the network heavily. Events for a given host typically have a small ra nge for $\boldsymbol{\Delta}$ (standard deviation ($\boldsymbol{\Delta}$) / mean ($\boldsymbol{\Delta}$) ~ 0.11 ± 0.1) indicating that the backup routes or diurnal behavior is consistent. This manifests itself in a multi-modal Distribution Function for $\boldsymbol{\Delta}$.

By careful examination of candidate events detected with $\boldsymbol{\delta = 0}$ (and ignoring whether the events are diurnal) we classify all candidates as to whether they are events we are interested in or not (i.e. exhibit sharp drop in bandwidth , persist for a long term (> > 3 hours) and are large enough). With $\boldsymbol{\delta}$ = 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 $\boldsymbol{\delta}$ to 33% we get 32% misses and 2% false positives. In this case 15% of the hits are diurnals.

If one eliminates the hosts with large diurnal variations in their bandwidth, then the plateau algorithm is quite successful in detecting step changes in bandwidth. However, this is a big "if", and our next step (to be reported on at the workshop) will be to incorporate filtering of the diurnal effects.

B. HW Algorithm

The current implementation of the bandwidth analysis system is undergoing continuous tests. The sensitivity of the applied algorithm is closely related to the $\mathbf{X^2}$ threshold value and to the completeness of the measurements. As can be seen in Fig. 4 there are 5 significant shifts in performance and all were identified. Also there are 4 lost observation events and correlated with them "drop" events. Only 2 false positive alerts could be eliminated by a higher $\mathbf{X^2}$ threshold. For very noisy data an additional criterion as Average($N_{time\text{-}window}$) < (Average($N_{time\text{-}window}$ *for "good"* $\mathbf{X^2}$)*0.95) could be applied to avoid false positive alerts. Also, wavelet [18] decomposition with appropriate threshold (5% variations) is possible to generate "clean er" statistics. There is no additional algorithm involved to separate anomalies by type of event, all anomalies are treated equally from the end -user point of v iew. The presented results are currently based only on one -way iperf tests. Due to the usage of normalized values the whole analysis could be applied to any type of monitoring statistics.

V. Work in Progress

The following studies are in progress and we plan to report on them in the final paper.

- We will extend the library of events and use it to quantify false positives and misses for the various algorithms;
- We will apply the plateau algorithm to iperf TCP achievable throughput measurements made at 90 minute intervals;
- We will extend the plateau technique to account for diurnal changes;
- We will apply the HW algorithm to the *Cap* measurements;
- We will apply the HW algorithm to the bi-directional iperf tests;
- We will research existing algorithms to choose optimal forecasting parameters for HW algorithm;
- We are developing a subspace Principal Components Analysis technique (PCA) [19] and will explore its use for our end-to-end metrics and paths;
- We are in contact with the developers of a neural network technique [20] for detecting events and if successful will compare that technique with the others;
- We will extend the implementations to also detect step up increases so we can apply to metrics such as RTT and also determine the anomaly duration.

The above studies will enable us to make an extensive comparison of three (possibly four) anomalous event detection techniques on a wide range of real end-to-end Internet performance metric measurements with a wide range of paths and frequency of measurements (minutes to hours).

VI. Conclusions

For measurements with limited diurnal (or other seasonal) changes the plateau algorithm technique works well, is easily understood by people with a non-statistical background and has easy to interpret user settable parameters. The effect of diurnal changes for our paths is sufficient, however, to make it necessary to incorporate the effect. The work of [14] indicates that the plateau algorithm also works well for RTT type ping measurements.

The HW technique explicitly incorporates seasonal changes and so should work better on paths with significant diurnal changes. It is sensitive to the choice of the forecasting parameters and the size of the moving time window. The choice of the initial parameters can be tricky. Additional study is needed to simplify this by applying a minimization algorithm, combined with ideal measurements for every node.

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. Our implementation of HW takes about 7 minutes on a ~ 1GHz Intel Xeon host to analyze about 43K *Cap* measurements (measurements are at ~ 3 minute intervals) while plateau takes about twice as long.

The subspace PCA analysis has been reported [20] 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*) and paths simultaneously. On the other hand it is less intelligible to someone without a statistical background.

Once we have a robust, reliable anomalous event detection technique we will use it generate alerts. These will be filtered relevant information gathered from network devices, analyzed and reported to network administrators.

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, Ruchi Gupta for her help in coding the plateau algorithm. We also thank Mark Crovella for useful discussions on the subspace PCA technique.

References

[1] McGregor A, Braun H-W, and Brown J, " The nlanr network analysis infrastructure", IE*EE 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] "Iperf the TCP/UDP Bandwidth Measurement Tool", available http://dast.nlanr.net/Projects/Iperf/
[10] "BBFTP Large Files Transfer Protocols", available at http://doc.in2p3.fr/bbftp/
[11] "Globus Grid FTP Protocol and Software", available at http://www.globus.org/datagrid/gridftp.html
[12] 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.
[13] Logg C, Cottrell R. L. and Navratil J., "Experiences in Traceroute and Available Bandwidth Change Analysis", *SIGGCOMM'04 Workshops* , Aug 30 & Sep 3, 2004, Portland OR, USA
[14] McGregor A.J. and Braun H-W, "Automated Event Detection for Active Measurement Systems", Proceedings of PAM2001, Amsterdam, Netherlands, April 2001
[15] Brockwell P. and Davis R, "Introduction to Time Series and Forecasting", *Springer* New York, 1996
[16] Brutlag J.D., "Aberrant Behaviour Detection in Time Series for Network Monitoring", Proceedings of LISA 2000, New Orleans, LA, USA, December 2000.
[17] NIST e-handbook of statistics, http://www.itl.nist.gov/div898/handbook.
[18] Barford P , Kline J , Plonka D, and Ron A , "A Signal Analysis of Network Traffic Anomalies", Proceedings of the second ACM SIGCOMM Workshop on Internet measurement, Marseille, France, 2002.
[19] Lakhina A, Crovella M, Diot C, "Diagnosing Network-Wide Traffic Anomalies", *Sigcomm* 2004.
[20] 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.