Experiences in Traceroute and Available Bandwidth Change Analysis[1]

Connie Logg and Les Cottrell
Stanford Linear Accelerator Center

ABSTRACT

SLAC has been studying end-to-end WAN bandwidth availability and achievability for 2.5 years via IEPM-BW[1]. IEPM-BW performs network intensive tests every 90 minutes. Based on that experience we have also developed a light weight available bandwidth (ABWE[2]) measurement tool which can make a measurement within a second. We are now extending this to a WAN measurement and detection system (IEPM-LITE) aimed at troubleshooting network performance problems. IEPM-LITE uses ping, forward and reverse traceroutes, and ABWE sensors to monitor, in real-time, changes in available bandwidth and routes to and from target hosts. This paper discusses the experiences, techniques and algorithms used to detect and report on significant traceroute and available bandwidth changes. The ultimate aim is to develop a lightweight WAN network performance monitoring system that can detect in real time significant changes and generate alerts.

keywords: bandwidth, availability, problem detection, traceroutes, real-time alerts, WAN, network monitoring

INTRODUCTION

Modern day High Energy Nuclear Physics (HENP) research and analysis requires that large volumes of data be effectively distributed to collaborators around the world. In September 2001, SLAC embarked on a WAN performance testing and analysis project known as IEPM-BW[1], to evaluate and monitor the performance of the paths to its collaborators. The IEPM-BW system is network intensive in its measurements. This was necessary in order to measure the performance in a way that mimicked the large volume data transfers. Concurrently, a lightweight, non-intensive bandwidth measurement tool (ABwE[2]) was being developed. The intent is that we be able to perform the same testing and analysis using non-network intensive tools, and on a more frequent basis to present real-time feedback on critical network links.

SLAC has been active in WAN monitoring for several years. One of the drawbacks of all this monitoring is that there are thousands of reports and graphs generated daily, which no one has time to look at. Critical performance changes often go unnoticed. For example, in August 2003[3], we did not notice until about a month later that IPERF throughput from SLAC to CALTECH had dropped dramatically. Once this drop was noticed and investigated, it was fixed in about 4 hours. If we had been alerted to this change, it could have been detected much faster, and fixed in a timely fashion.

The system we are now testing is referred to as IEPM-LITE. It performs pings, forward and reverse traceroutes, and frequent light-weight available bandwidth measurements using ABwE [2]. This paper discusses the techniques for bandwidth change detection and traceroute analysis which we are currently developing to generate alerts, and when possible, correlate available bandwidth changes to route changes. Ping and traceroute information is gathered every 10 minutes. Available bandwidth measurements are made every minute.

AVAILABLE BANDWIDTH CHANGE DETECTION

The principle behind this algorithm is evolved from work by NLANR [4]. It involves buffering the time sequence ABwE data into two buffers: a history buffer (HISTBUF) for base-lining, and when a datum meets specific requirements, into a trigger buffer (TRIGBUF). Each buffer has a maximum number of entries parameter, HISTMAX and TRIGDUR respectively. TRIGDUR determines how long the bandwidth change must exist before an analysis of its data is performed to see if we have encountered an alert condition. Note that since the ABwE data is taken once a minute, TRIGDUR is the number of

[1] This work is supported by U.S. DOE Contract No. DE-AC03-76SF00515

minutes (assuming no data is lost) that a drop must exist before an alert is raised. The history buffer also has a HISTMIN parameter, which is only used at startup to "prime" the history buffer. HISTMIN is the number of data points which are added to the history buffer when the analysis starts up.

Once HISTBUF is primed with HISTMIN data points, we are ready to enter the data processing loop. The mean (HISTMEAN) and standard deviation (HISTSD) of the HISTBUF data are calculated, and used as a reference in the analysis.

There are several critical parameters which are used in the analysis and alert generation. Currently these must be manually tuned, as they are dependent on the size of change and length of time a bandwidth depression is to be present before an alert is generated. At some point in the future, we hope to be able to auto-configure some of these parameters by pre-examining the data. Others such as the buffer lengths will probably depend on the users' requirements. For example how long must a change be sustained (TRIGDUR) before it is considered significant depends on how long the user wants a drop to be sustained before he is notified.

- Sensitivity (SENS) – the number of standard deviations beyond the HISTBUF mean which a datum must lie to be considered a trigger value. The default at this time is 2, however we are evaluating how to dynamically set this as a function of HISTMEAN and HISTSD.
- Threshold (THRESH) – is the % difference between the HISTBUF mean and the TRIGBUF mean which must be passed for an alert to be generated. Once we are in an alert state, this threshold must again be met before another alert is generated. We are in an alert state when an alert has been issued and we have not seen data that is not trigger data. As soon as non-trigger data is encountered, the alert state variable ALMEAN is zeroed.

The SENS parameter is used in 2 tests which are applied to the data. Note although the algorithm is symmetric for bandwidth rises as well as drops, we only consider drops at this point.

- QTRIGGER – selects whether a data point should be put into the history buffer or the trigger buffer.
  QTRIGGER = TRUE if:
  Datum <= HISTMEAN – SENS*HISTSD
  Otherwise QTRIGGER = FALSE
- QOUTLIER – evaluates the datum to see if it is so far off the rest of the data that is should be considered an anomaly and not included in the mean and standard deviation calculations.
  QOUTLIER = TRUE if:
  DATUM>=HISTMEAN+2*SENS*HISTSD
  Otherwise QOUTLIER = FALSE.

HISTBUF is primed and HISTMEAN and HISTSD are calculated. The analysis loop uses the following logic to process the remainder of the data.

1. TOP OF LOOP – Read in the next data point
2. If it is <= 0 (invalid data check), skip it and go to TOP OF LOOP
3. Is it a TRIGGER value? NO:
   - Is it an OUTLIER? – NO: Add it to HISTBUF and recalculate HISTMEAN and HISTSD
   - Is it an OUTLIER? – YES: ignore the data point
   - In either case, drop the oldest value from TRIGBUF so that we age out the trigger buffer data in case the bandwidth drop under examination terminates.
   - If TRIGBUF is empty, ALMEAN=0. That is, terminate any existing alert state.
   - Go to TOP OF LOOP
4. Is it a TRIGGER value? – YES
   - Add it to TRIGBUF
   - Is TRIGBUF full? NO: go to TOP OF LOOP
5. TRIGBUF is full: Check for an ALERT
   - Calculate the mean (TRMEAN) of the data in TRIGBUF
   - Calculate the percent change from HISTMEAN:
   - %change =
     (HISTMEAN-TRMEAN) /HISTMEAN
   - If %change < THRESH, remove the oldest value from TRIGBUF (This is not put into HISTBUF, because if the drop is gradual, we may not detect the drop). Go to TOP OF LOOP.
   - If %change > THRESH, and we are NOT in an alert state (ALMEAN=0), issue the alert, load TRIGBUF into HISTBUF, and recalculate HISTMEAN and HISTSD. Save TRMEAN into ALMEAN so that we know we are in an alert state. Go to TOP OF LOOP.
   - If %change > THRESH, and we ARE in an alert state, calculate
     %trchange=
     (ALMEAN-TRMEAN)/ALMEAN
     If %trchange > THRESH (we have dropped more than the major threshold again), issue another alert, load TRIGBUF into HISTBUF, and recalculate HISTMEAN and HISTSD. Save the TRMEAN in ALMEAN so we know the new alert state. Go to TOP OF LOOP.

COMMENTARY AND EXAMPLES

If there are breaks in the data, they are ignored, and the analysis continues.

Let's look at how the sensitivity, threshold, and trigger buffer affect the algorithm.

Given reasonable values for the sensitivity and threshold, the trigger buffer length has the most dramatic effect on the frequency of alerts. It determines the amount of time bandwidth must be depressed in order to check for an alert. In Figures 1 & 2, we can see the effect of trigger buffer length. In the figures, the asterisks on the top axis indicate that an alert occurred at that point in time.

In Figure 1, the TRIGDUR = 30, that is, since the data points are 1 minute apart, the bandwidth must be depressed for 30 minutes before an alert is issued.

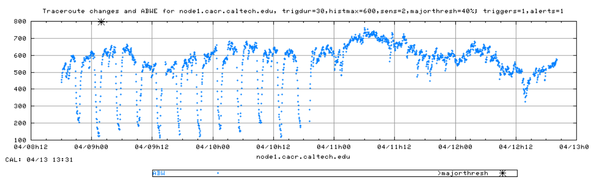


Figure 1: Trigger buffer length is 30 – one alert

In Figure 2, the trigger buffer length is 10, and so we have more alerts since the amount of time that the bandwidth must be depressed is less.

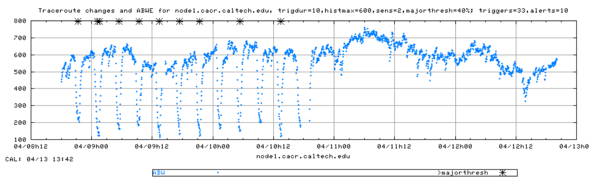


Figure 2: Trigger buffer length is 10 – nine alerts

The sensitivity controls how far off the mean a data value must be to be entered into the trigger buffer. Although the sensitivity is mitigated by the trigger buffer length, as the sensitivity increases, the number of alerts often increases, as only the values representing the more dramatic drops are put in the trigger buffer, and hence the threshold of change is reached more often.

The threshold is the percent change from the norm or previous alert that is required before an alert is issued. 40% has been chosen here for the threshold. The larger the threshold, the fewer the alerts, however it needs to be set to represent a point of reasonable concern over a range of bandwidth values. A threshold of 10% for a gigabit link, or a hundred megabit link might be passed simply because of an increase of the cross traffic. This can happen frequently. We are only concerned about being alerted for dramatic bandwidth changes that indicate the possibility of potential throughput problems on a link. We would like to know before the throughput drops 50%-60%, and thus chose 40% as the threshold.

While initially implementing the algorithm of [4] (which was applied to ping RTTs) to analyze our available bandwidth drops, we discovered that various characteristics in our available bandwidth data required many changes to account for the wide variability in the data. The algorithm described in [4], uses the variance to separate "trigger" and "outliers". The large variability in our data made use of the variance impractical.

Another change we made was a result of our observations that bandwidth can drop gradually over a long period of time. Data that may be flushed from the trigger buffer when a full trigger buffer is not an alert can gradually lower the mean of the history buffer, if it is added to the history buffer. Therefore, data dropped from the trigger buffer as an examination shows that no alert is warranted, is discarded.

TRACEROUTE ANALYSIS

We currently utilize the standard Linux traceroute with a 2 second timeout, 1 probe per hop, a maximum hop count of 30 hops, and start the traceroute after leaving the SLAC border. For each destination host we study the performance of traceroute with UDP and ICMP probes and choose the most appropriate probe protocol (i.e. resolves the route before the 30 hop maximum and/or minimizes the number of non-responding routers). By default we use UDP probes.

The goal of the traceroute analysis is to categorize the traceroute information and detect "significant route changes" between the current traceroute and one taken previously. The algorithm for categorizing the traceroutes is conceptually as follows. For each hop of a traceroute we compare the router information (IP address) against the router information for the same hop for the previous traceroute measurement for a given path. If the router for this hop did "not respond" (i.e. the traceroute reported an asterisk (*)) for either this or the previous traceroute then the Hop Change Information (HCI) for this hop is noted as "unknown". If the router responded (i.e. provided an

IP address) for this hop, for both this and the previous traceroute, then the IP addresses reported for this hop are compared:

- If they are identical then the HCI is marked as "no change".
- If the addresses are not identical then:
  - if they only differ in the last octet then the HCI is marked as "minor change same 3rd octet".
  - if the addresses are in the same Autonomous System (AS) then the HCI is marked as "minor change same AS".
- If neither "minor change same 3rdx octet" nor "minor change same AS" are identified then the HCI is marked as a "significant route change". We also sub-classify the "significant route change" into whether or not the change involves one ("minor significant route change") or more hops ("major significant route change").

When all the hops have been compared between the current and previous traceroutes, then precedence is given to any "significant route change", followed by "minor change same AS", "minor change same subnet", "unknown", and "no change" in that order.

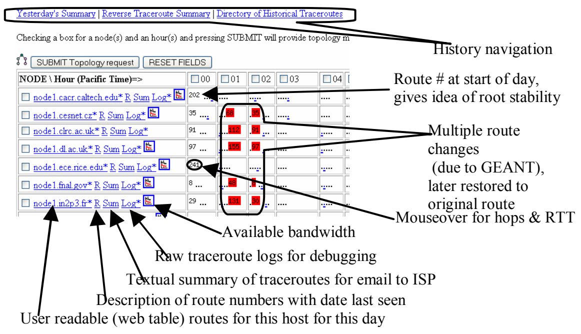
In addition, unless the HCI is set to "no change" we also note whether the current traceroute did not terminate until the "30 hop" limit was reached and/or whether the destination is pingable. Since the destination hosts are chosen to be normally pingable, a non pingable destination usually means the destination host or site is not reachable, whereas a "30 hop" pingable destination is probably hidden behind a firewall that blocks traceroute probes or responses.

In all cases except "significant route changes" we also note whether an ICMP checksum error was reported in a current traceroute.

One other case that is noted is the traceroute reporting "host unknown" which probably means the host name is currently not resolvable.

DISPLAYING TRACEROUTE INFORMATION

The information is displayed in a table representing the routes for a single day (Figure 3). The table columns represent the hour of the day and each row represents a remote destination host. The rows are labeled with anonymized host names and URL links



Yesterday's Summary | Reverse Traceroute Summary | Directory of Historical Traceroutes

Checking a box for a node(s) and an hour(s) and pressing SUBMIT will provide topology m

NODE \ Hour (Pacific Time)=>	☐00	☐01	☐02	☐03	☐04
☐ node1.cacr.caltech.edu* R Sum Log*	202	....	....	....	....
☐ node1.cesnet.cz* R Sum Log*	35	..68	35 ..	....	....
☐ node1.clrc.ac.uk* R Sum Log*	91 ...	..112	91 ..	....	....
☐ node1.dl.ac.uk* R Sum Log*	97 ...	..155	97	....	....
☐ node1.ece.rice.edu* R Sum Log*	241	....	....	....	....
☐ node1.fnal.gov* R Sum Log*	8 ...	..48	8 ..	....	....
☐ node1.in2p3.fr* R Sum Log*	29 ...	..131	30 ..	....	....

SUBMIT Topology request
RESET FIELDS
History navigation
Route # at start of day, gives idea of root stability
Multiple route changes (due to GEANT), later restored to original route
Mouseover for hops & RTT
Available bandwidth
Raw traceroute logs for debugging
Textual summary of traceroutes for email to ISP
Description of route numbers with date last seen
User readable (web table) routes for this host for this day

Figure 3: Section of a Traceroute Summary Table

are provided to: an HTML table of the day's routes, a text table of the routes, route number information (i.e. the route number, the associated route and the time last seen), the raw traceroute data, plots of the available bandwidth alert analysis information and the ABwE dynamic bandwidth capacity, cross traffic and available bandwidth. The columns are labeled with the hour of the day.

For each non "significant route change" the cells of the table contain a single colored character for each traceroute measured in that hour. The single character represents the HCI or "30 hop" route categorization (i.e. period = "no change", asterisk = "not respond", colon = "minor change same 3rd octet", a="minor change same AS", vertical bar = "30 hop", exclamation mark = "host unknown") and the characters are colored orange if an ICMP checksum is noted, red if the destination host is not pingable, and black otherwise. The use of a single character to display the route categories allows a very dense table to be displayed which in turn facilitates visually scanning for correlated route changes occurring at particular times for multiple hosts and/or hosts that are experiencing multiple route changes in a day.

For each "significant route change" the route number is displayed colored red if changes were noted in more than one hop, and orange otherwise.

If an available bandwidth alert was noted for a destination in an hour then that cell's background is colored to indicate the existence of the alert.

Each row and each column also contains a check box (Figure 4) that can be selected to submit requests for either a topology map for the selected hosts and times, or the routes together with their router AS information.

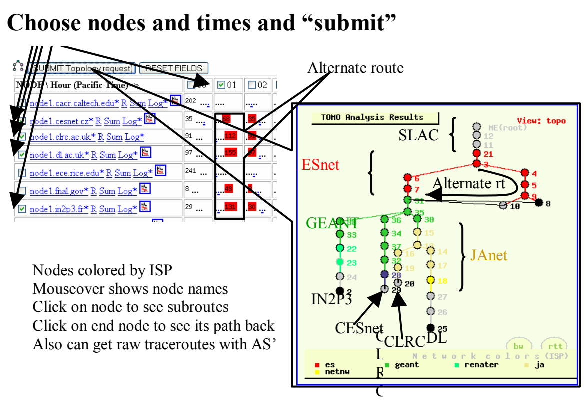

Choose nodes and times and "submit"
Alternate route
TOMO Analysis Results
View: topo
SLAC
ESnet
Alternate rt
GEANT
JAnet
IN2P3
CESnet
CLRC
DL
Nodes colored by ISP
Mouseover shows node names
Click on node to see subroutes
Click on end node to see its path back
Also can get raw traceroutes with AS'

Figure 4: Traceroute display selected by check boxes on Traceroute Summary Table

When the "Submit topology request" is clicked, the traceroute information is turned into a graphical display.

The web page also includes documentation and provides access to the reverse traceroutes, and historical data.

IN SUMMARY

We have described analysis techniques for examining bandwidth and traceroute changes independently. Our intent however, is to marry the two into displays which integrate the information available from both of them. One such display is Figure 5 in which solid vertical lines indicate when traceroute changes occur, and the graphic displays the available bandwidth and alerts due to significant changes in the available bandwidth. Another is shown is Figure 6, where the primary display is the traceroute summary page, and the bandwidth changes are indicated by coloring the hour boxes for a node where changes have been detected. Note the "node1.cacr.caltech.edu" line, and, look again at Figure 2.

FUTURE WORK

Areas for future exploration include further study and development of the bandwidth change algorithm (including accommodating diurnal and other periodic changes), and providing a better understanding of how to set the algorithm's parameters to meet required needs. We intend to apply the bandwidth change algorithm to other data including ABwE RTT and ping RTT estimates, and extend the tools to analyze AMP [5] data. We are also exploring a more efficient traceroute that does the probing in parallel.

Other work will include implementing the bandwidth change detection algorithm in a symmetric fashion so that we can detect rises in throughput and thus be able to quantify the time span of varying throughput levels.

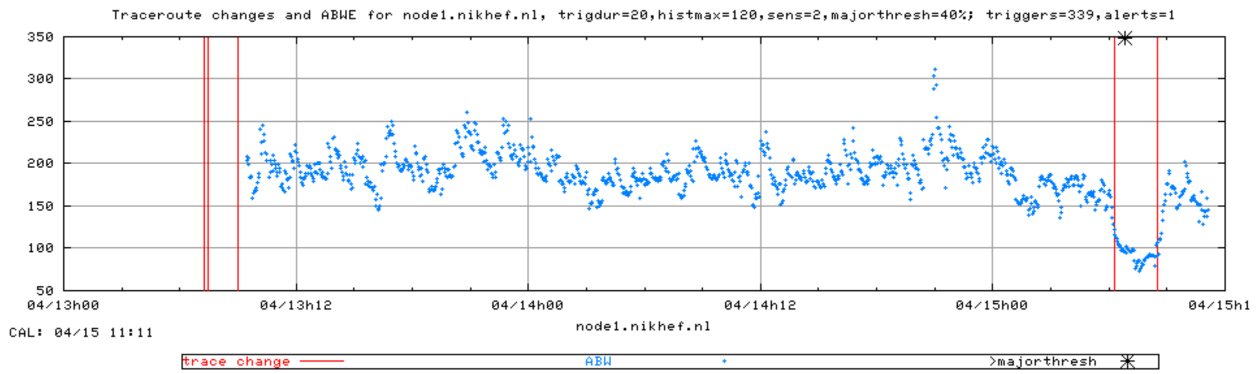
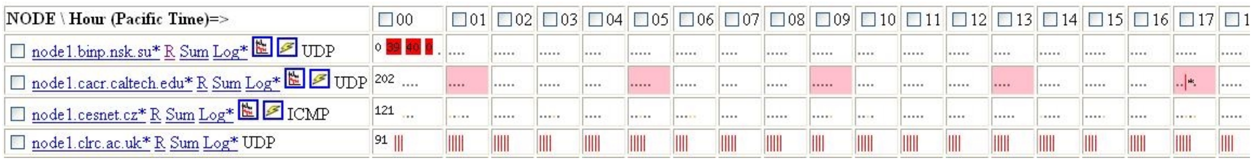


Figure 5: Traceroute changes Integrated with Available Bandwidth Analysis and Display



NODE \ Hour (Pacific Time)=>	☐00	☐01	☐02	☐03	☐04	☐05	☐06	☐07	☐08	☐09	☐10	☐11	☐12	☐13	☐14	☐15	☐16	☐17	☐1
☐ node1.binp.nsk.su* R Sum Log* UDP	0 39 40 0 .	.....	.....	.....	....	.....	.....	....	.....	.....	....	.....	.....	....	.....	.....	....	.....	.....
☐ node1.cacr.caltech.edu* R Sum Log* UDP	202	....	.....	.....	....	.....	.....	....	.....	.....	....	.....	.....	....	.....	.....	....	..|*.	.....
☐ node1.cesnet.cz* R Sum Log* ICMP	121 ...	.....	.....	.....	....	.....	.....	....	.....	.....	....	.....	....	.....	.....	.....	....	.....	.....
☐ node1.clrc.ac.uk* R Sum Log* UDP	91 ||	|||||	|||||	||||	|||||	|||||	||||	|||||	|||||	||||	|||||	|||||	||||	|||||	|||||	||||	|||||	|||||	||||

Figure 6: Bandwidth Changes integrated with the Traceroute Summary Display

REFERENCES:

[1] *Experiences and Results from a New High Performance Network and Application Monitoring Toolkit*, Les Cottrell, Connie Logg and I-Heng Mei, SLAC-PUB-9641, presented at PAM 2003.
[2] *ABwE: A practical Approach to Available Bandwidth Estimation*, Jiri Navratil and Les Cottrell, SLAC-PUB-9622, presented at PAM 2003.
[3] Case study of August 2003 bandwidth drop between SLAC and CALTECH: http://www.slac.stanford.edu/grp/scs/net/case/caltech
[4] *Automated Event Detection for Active Measurement Systems*, A. J. McGregor and H-W. Braun, Passive and Active Measurements 2001. http://byerley.cs.waikato.ac.nz/~tonym/papers/event.pdf
[5] *The NLANR NAI Network Analysis Infrastructure,* McGregor A., Braun H-W. and Brown J. IEEE Communication Magazine special issue on network measurement, pp122-128, May 2000.