# Analyzing, Identifying & Alerting on Network Issues

*Petya* Vasileva[1,4], *Marian* Babik[2], *Shawn* McKee[1], and *Ilija* Vukotic[3]

[1]University of Michigan Physics, Ann Arbor, MI, USA
[2]European Organization for Nuclear Research (CERN), Geneva, Switzerland
[3]Physics Department, University of Chicago, Chicago, IL, USA
[4]Faculty of Mathematics and Informatics, University of Plovdiv, Bulgaria

**Abstract.** The Worldwide LHC Computing Grid (WLCG) relies on the network as a critical part of its infrastructure and therefore needs to guarantee effective network usage and prompt detection and resolution of any network issues, including connection failures, congestion, and traffic routing. In this paper, we will describe our ongoing work to proactively analyze, correlate and alert on various network and infrastructure issues. We will discuss the methods and techniques applied, the systems developed, and the challenges with the measurements that make it difficult to easily identify problems or assign those problems to the appropriate location(s).

## 1 Introduction

The Open Science Grid (OSG)[1] and the Worldwide LHC Computing Grid (WLCG)[2] together form a complex network of computer systems. These grids interconnect sites around the world, supporting multiple scientific experiments and facilitating the daily transfer of substantial data between their storage systems. Unfortunately, network failures, bottlenecks, and other issues are inevitable and often difficult to detect. In addition, these network issues can significantly impede scientists in their research endeavors.

To address these challenges, the Service Analysis and Network Diagnostics (SAND)[3] team has established a global network of perfSONAR[4] measurement agents (toolkits), which track end-to-end network paths, assess network performance, and centrally store this wealth of information in an Elasticsearch (ES) database [5]. These toolkits continuously monitor latency and packet loss across more than 6000 combinations of source-destination pairs, spanning international research and education networks (R&EN). Additionally, the project tracks throughput for a similar number of source-destination pairs, capturing data on timescales ranging from hours to days. Furthermore, traceroute measurements are conducted between all pairs at intervals of 10-60 minutes. This expansive deployment evolved to support the monitoring of the Worldwide LHC Computing Grid (WLCG) sites, which currently count approximately 250 instances [6]. Notably, perfSONAR toolkits are strategically co-located with the main site storage infrastructure, ensuring that the resulting network measurements reflect the site paths utilized by data transfers.

The data pipeline has been operational since 2018, and Table 1 illustrates the volume and frequency of data collected over a three-year period (2019-2023). The data set represents

a unique and exceptionally thorough collection of metrics with the potential to reveal fresh perspectives on the dynamics and attributes of global research and education networks. The data from the network pipeline is stored in the ES cluster at the University of Chicago. Its voluminous and diverse nature, coupled with its high data rate, presents challenges when proactively looking for any kind of issues.

| Data Type | Total tests | Tests/Day | Stored size |
|---|---|---|---|
| Latency | 11.4 B | 2.67 M | 4.11 TB |
| Packet loss | 11.6 B | 2.67 M | 3.16 TB |
| Throughput | 23.1 M | 2.02 K | 8.83 GB |
| Traceroute | 3.98 B | 1.11 M | 3.25 TB |

**Figure 1:** The type, frequency and volume of data gathered by the perfSONAR toolkits in 2019-2023.

In light of this extensive data set and its inherent challenges, efficient analytics plays a pivotal role in promptly detecting and alerting to network issues, while user-friendly data visualizations facilitate understanding of diverse problems. Consequently, we introduce two applications: the Alarms and Alerts Service (AAAS) [7] and pSDash [8], where the first generates alarms and the second provides visualizations. These tools enable users and administrators to get a problem-oriented view of their sites in the broader global network context. Moreover, they are instrumental in assessing test configurations and facilitating notifications for specific alarms for individual sites.

## 2 Measurement Platform and Associated Data

This work is entirely based on the measurements performed with the perfSONAR toolkits. The data for each link (or a source-destination pair) are collected at a specific rate and measure a variety of aspects representing the underlying network. The primary focus is on the latency, throughput, and traceroute tests.

The **latency** measures the time it takes for the data to travel between the two hosts using the one-way active measurement protocol (OWAMP)[9]. It involves sending timed packets at a rate of 10 Hz (by default). Additionally, this latency measurement inherently provides insight into packet loss, as it counts how many of the expected 600 packets arrive at the destination each minute.

PerfSONAR typically measures **throughput** using the performance of a single TCP flow. Through iperf3, it performs memory-to-memory transfer tests using UDP and TCP and provides reports on TCP retransmissions and the size of the congestion window. These tests are run on a scale between 6 and 24 hours.

The **traceroute** test is used to discover the network path between two toolkits throughout the network. This process aims to identify all layer 3 routers along the path. Measurements are executed in a 10-minute interval between all latency and throughput endpoints.

The test result is then stored in a measurement archive for further analysis. Kibana [10] dashboards provide various instruments for visualization; however, the complexity and size of the data sets require a higher level of processing in order to understand the dynamics of the

network. To effectively extract valuable insights, it is essential to analyze the entire range of diverse metrics we have collected, with a particular focus on understanding how these metrics evolve over time. Moreover, the toolkits test more than 6000 pairs, mostly located at different institutes across the world, and thus have a unique set of configurations and capacities. These challenges were considered and, as a result, two new tools were developed: AAAS and pSDash.

## 3 Platform Component Description

AAAS provides a simple way to store alarms, handle user subscriptions, and create alerts. Alarms are logically grouped into types and are generated periodically. A RESTful Application Programming Interface (REST API) enables the storage of novel alarms. The users can specify a period, tag sites, and select from a list of multiple types of alarms on which to be alerted. Once subscribed, users receive an email containing the alarms of interest as well as the URLs pointing to pSDash for further details.

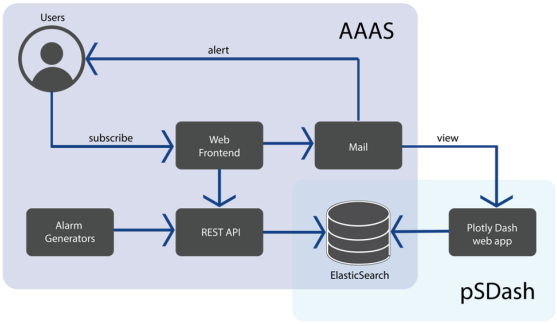


**Figure 2:** As shown above, AAAS goes through a cycle of subscription, generation and storage of alarms, and e-mailing alerts to users, while pSDash queries the data from ES and provides visualizations about the problems.

pSDash is a web application based on Plotly.js [11] and Plotly Dash [12]. Provides interactive visualizations of network problems and lets users search for alarms. Screenshots that illustrate the application are presented in Figures 3, 4, 5, 6. pSDash is structured so that new features and/or pages can be easily added. The underlying codebase is organized into a collection of classes, responsible for caching recent data within project-local files, and a set of individual files, each encapsulating the entire logic of a single page.

## 4 Implementation of Problem Detection and Its Future

Several Python scripts[7] are the basis for generating alarms. They are scheduled to run at least once a day, depending on the type and granularity of the alarms. We have introduced some hard limits on most of the detected problems in order to capture the most significant issues. The description and the current list of alarms are as follows:

1. Based on throughput data. Bandwidth measures are irregular for many pairs. Thus, the alarm considers a much broader period (i.e. 21 days) to identify what is a typical throughput for a given link.

- **Bandwidth decreased** (from/to multiple sites) is an alarm that indicates a major drop in throughput for a link (see Figure 3). The procedure captures single- and multisite bandwidth degradation. The latter is a major event where a site experiences issues in both directions - as a source and as a destination of measures, or in a single direction - from or to multiple other sites;

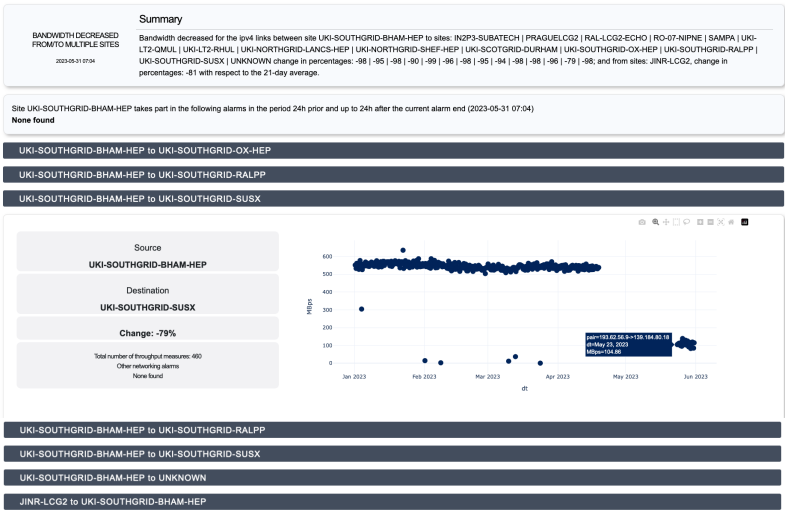


**Figure 3:** Severe bandwidth decreased multi-site alarm on UKI-SOUTHGRID-BHAM-HEP, detecting throughput issues in both directions (from and to site)

- **Bandwidth increased** (from/to multiple sites) is the opposite of "Bandwidth decreased" - it logs the significant increase of throughput values. This is the only alarm that does not designate a problem but rather an improvement. It is helpful for monitoring long-term performance issues.

2. Based on latency/one-way delay data. These alarms track the time it takes packets to get from a source to a destination. Well-synchronized clocks are crucial for accurate measurement.

   - **Bad one-way delay measurements** is generated if a node reports time greater than 100ms;
   - **Large clock correction** alarm calculates clock corrections for all nodes that appear as both source and destination. An alarm is generated if the calculated value is greater than 100ms;

3. Based on packet loss data. Alarms consider the percentage of lost packets for every link.

   - **Complete packet loss** alarm is created when a link drops all packets;
   - **Firewall issue** is an alarm generated when node is involved in links that lost 100% of its packets for all tests in a given period or when the number of links (having lost all packets) is more than 10;
   - **High packet loss** problem alerts for packet loss above 2%. Similarly to the "Bandwidth decreased" alarms, these alarms have two variations - single alarm on a partic-

ular link; or multi-alarm (see Figure 4)- when one endpoint reports issues on more than five other endpoints, regardless of its position (as a source or as a destination);


**Figure 4:** High packet loss multi-site alarm on FZK-LCG2, showing issues in both directions (from and to the site)

4. Based on Traceroute data. Tracking changes on the path is not trivial due to the dynamic nature of the network.

   - **Path changed** alarm relies on the ASNs (Autonomous System Numbers) to reduce the complexity of a network built on top of thousands of IP addresses. Figure 6 describes the approach we used to identify divergence from the baseline path, which is often the most taken. When a new ASN appears, there is an alarm that logs all affected sites, while pSDash provides various graphs with respect to the change (see Figures 5 and 6);

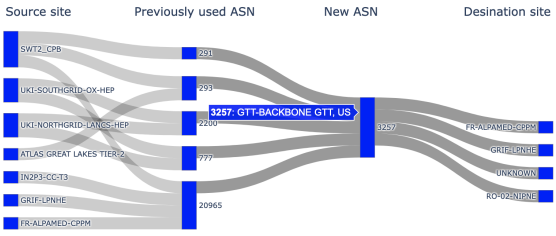


**Figure 5:** Visualization of the affect of a changed path. When a new ASN appears (or disappears), the topology changes for many links and possibly affects the overall performance on the file transfers between sites.


**Figure 6:** Left: Three distinct paths were reported. All AS numbers on the first alternative path participate in the baseline path. The second alternative path includes 3257 which is not on baseline; Right: AS numbers for every hop and the frequency of their occurrences at each position (source and destination not included). The dark blue values of 1 mean that the ASN was always used at that position; Close to 0, indicates that the ASN rarely appeared; OFF represents that the device did not respond at the time of the traceroute test; 0 is when there was a response, but the ASN was unknown.

- **Destination cannot be reached** (from any/multiple hosts). As the name suggests, the alarm tracks all nodes that block tests such that the destination is never the last value on the list of hops;
- **Source cannot reach** (any/multiple hosts) is a problem reported when a single source host cannot reach any or multiple destinations;

Key role into understanding the R&EN networks is to be able to correlate network metrics. However, each perfSONAR test category has different execution times. To overcome that obstacle, recent developments attempt to combine traceroute data with bandwidth values. This approach uses the trace paths (just before and just after the throughput test) as valid paths and attributes the bandwidth to both.

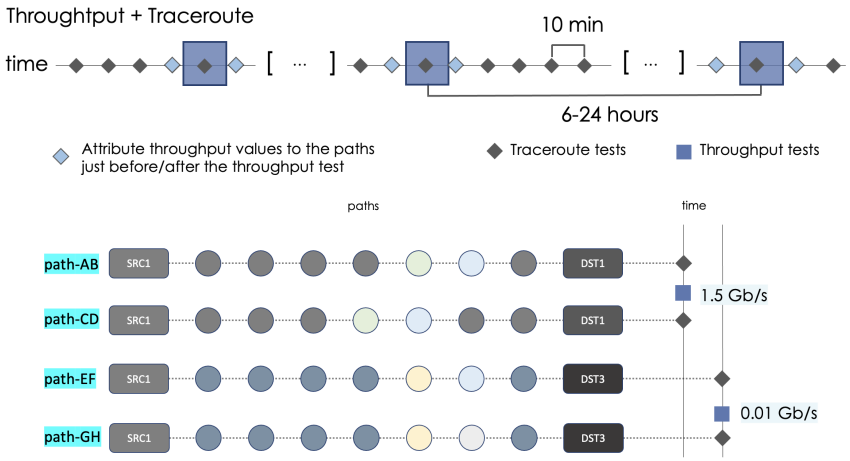


**Figure 7:** Discard all traceroute results except for the two around a throughput measure.

As seen in Figure 7, the method attaches one bandwidth value to all routers on the two paths, before the test started and after it finished. The result is a vector of throughput values for every router. Finally, we look for a downward trend, considering the number of values, number of paths, and the maximum throughput seen on the paths that pass through the router (see Figure 8). The described method provides a good estimate of the location of a network problem and could serve as a new type of alarm as well as a starting point for debugging ongoing issues.

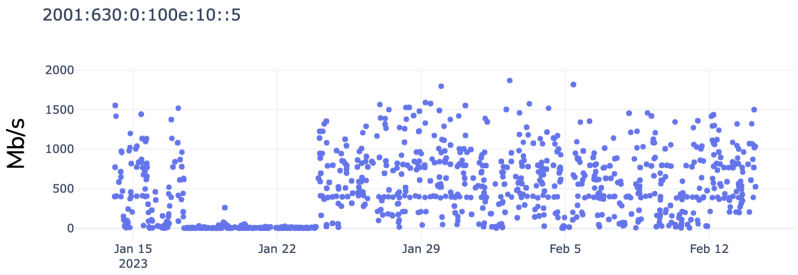


**Figure 8:** The throughput values attributed to a single router. This example clearly identifies an issue with or around that router.

The next step is to develop a procedure that automates the discovery of similar issues. In addition, we would like to prioritize alarms related to those links that are more often involved in the largest file transfers. And finally, to provide users with a way to acknowledge (pause) network alerts while working on resolving them.

## 5 Machine Learning Options and Plans

Our primary objective is to improve our understanding of network behaviors and fix faults as quickly as possible. However, as already discussed in the previous sections, the variability of network metrics requires new tools to be used in the localization of problems. We are currently working on integrating Machine Learning and Neural Networks based approaches into our systems and applications. In particular, we are researching different models that are capable of preserving the topology of the network and revealing cascading problems, faulty nodes, overloaded paths, and other issues that can be correlated.

## 6 Conclusion

With the upcoming upgrade to High Luminosity(HL) LHC [13], data volumes are projected to increase by a factor of ten to the exabyte scale ($10^{18}$ bytes) annually [14]. Thousands of scientists worldwide conduct data analysis and exchange of datasets relevant to their research. However, the resources of the network, including capacity, are finite and must be shared among these users [15]. Consequently, it becomes imperative to minimize congestion, address faults, and mitigate unexpected network behaviors.

We have demonstrated that the two systems described in this paper address these objectives and make progress towards a better understanding of the scientific network.

## 7 Acknowledgements

We acknowledge our collaborations with the CERN IT, WLCG and LHCONE/LHCOPN communities, who participated in this effort. We want to explicitly acknowledge the support of the National Science Foundation which provided long-term support for this work via:

OSG: NSF MPS-1148698, NSF SAND Project: NSF OAC-1827116, IRIS-HEP: NSF OAC-1836650.

## References

[1] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein et al. (2007), Vol. 78, p. 012057, http://stacks.iop.org/1742-6596/78/i=1/a=012057

[2] I. Bird, P. Buncic, F. Carminati, M. Cattaneo, P. Clarke, I. Fisk, M. Girone, J. Harvey, B. Kersevan, P. Mato et al., Tech. Rep. CERN-LHCC-2014-014. LCG-TDR-002 (2014), http://cds.cern.ch/record/1695401

[3] D. Weitzel, S. McKee, B. Bockelman, J. Thiltges, M. Babik, I. Vukotic, *The Service Analysis and Network Diagnosis Data Pipeline*, in *2021 IEEE Workshop on Innovating the Network for Data-Intensive Science (INDIS)* (IEEE Computer Society, Los Alamitos, CA, USA, 2021), pp. 1–11, https://doi.ieeecomputersociety.org/10.1109/INDIS54524.2021.00006

[4] A. Hanemann, J.W. Boote, E.L. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, D.M. Swany, S. Trocha, J. Zurawski, *PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring*, in *Service-Oriented Computing - ICSOC 2005*, edited by B. Benatallah, F. Casati, P. Traverso (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 241–254, ISBN 978-3-540-32294-8

[5] C. Gormley, Z. Tong, *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine* (" O'Reilly Media, Inc.", 2015)

[6] M. Babik, S. McKee, P. Andrade, B.P. Bockelman, R.W. Gardner, E.M.F. Hernandez, E. Martelli, I. Vukotic, D. Weitzel, M. Zvada, CoRR **abs/2007.00598** (2020), 2007.00598

[7] I. Vukotic, P. Vasileva, T. Shearer, (2023), *Alarms and Alerts*, Retrieved from https://github.com/sand-ci/AlarmsAndAlerts

[8] P. Vasileva, I. Vukotic, *An interface for the alarms of the osg/wlcg network measurements*, https://ps-dash.uc.ssl-hep.org/

[9] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, M. Zekauskas, RFC 4656, RFC Editor (2006)

[10] T.S. Project, *The OSG network metrics in ELK*, https://atlas-kibana.mwt2.org:5601/s/networking

[11] P.T. Inc., (2015), *Collaborative data science*, Retrieved from https://plot.ly

[12] P.T. Inc., *Dash - low-code framework for rapidly building data apps in python*, https://dash.plotly.com/

[13] G. Apollinari, O. Brüning, T. Nakamoto, L. Rossi, *High luminosity Large Hadron Collider HL-LHC* (2015), https://cds.cern.ch/record/2120673

[14] J. Zurawski, B. Brown, D. Carder, E. Colby, E. Dart, K. Miller, A. Patwa, K. Robinson, L. Rotman, A. Wiedlea, *High Energy Physics Network Requirements Review (Final Report, July-October 2020)* (2021), https://www.osti.gov/biblio/1804717

[15] E. Martelli, S. Stancu, *LHCOPN and LHCONE: Status and Future Evolution* (2015), Vol. 664, p. 052025, http://stacks.iop.org/1742-6596/664/i=5/a=052025