Review

International Journal of Modern Physics A
Vol. 36, No. 5 (2021) 2130005 (13 pages)

DOI: 10.1142/S0217751X21300052

World Scientific
www.worldscientific.com

TRACER (TRACe route ExploRer): A tool to explore OSG/WLCG network route topologies

Evgeniy Tretyakov* and Alexey Artamonov†

National Research Nuclear University MEPhI,
31 Kashirskoe shosse, Moscow, Russia
Plekhanov Russian University of Economics,
36 Stremyanny Lane, Moscow, Russia
**estretyakov@mephi.ru*
†*AAArtamonov@mephi.ru*

Maria Grigorieva‡

Lomonosov Moscow State University,
Leninskie Gory 1, bld. 4, Moscow, Russia
Plekhanov Russian University of Economics,
36 Stremyanny Lane, Moscow, Russia
maria.grigorieva@cern.ch

Alexei Klimentov

Brookhaven National Laboratory, Upton, New York, USA
aak@bnl.gov

Shawn McKee

University of Michigan, 450 Church St, Ann Arbor, Michigan, USA
smckee@umich.edu

Ilija Vukotic

University of Chicago, Chicago, Illinois, USA
ivukotic@uchicago.edu

Received 28 July 2020
Revised 17 November 2020
Accepted 21 December 2020
Published 10 March 2021

The experiments at the Large Hadron Collider (LHC) rely upon a complex distributed computing infrastructure (WLCG) consisting of hundreds of individual sites worldwide at universities and national laboratories, providing about half a billion computing job slots and an exabyte of storage interconnected through high speed networks. Wide Area

‡Corresponding author.

Networking (WAN) is one of the three pillars (together with computational resources and storage) of LHC computing. More than 5 PB/day are transferred between WLCG sites. Monitoring is one of the crucial components of WAN and experiments operations. In the past years all experiments have invested significant effort to improve monitoring and integrate networking information with data management and workload management systems. All WLCG sites are equipped with perfSONAR servers to collect a wide range of network metrics. We will present the latest development to provide the 3D force directed graph visualization for data collected by perfSONAR. The visualization package allows site admins, network engineers, scientists and network researchers to better understand the topology of our Research and Education networks and it provides the ability to identify nonreliable or/and nonoptimal network paths, such as those with routing loops or rapidly changing routes.

Keywords: Networks; visualization; perfSONAR; traceroute; topology.

PACS number: 07.05.Bx

1. Introduction

Computer networks are extremely important for data-intensive science, enabling the construction of complex distributed computing infrastructures that allow scientists to access data and computing resources locally, regionally, nationally and internationally. Network technologies enable collaborative research and data mobility and access around the world. Because of the foundational role that networks play in modern science, it is critical we ensure their high reliability and stability, but this is often very challenging. Networks by their nature cross multiple administrative domains, meaning that there is no single owner or manager of a given end-to-end network path. When soft failures (degradation of performance but not loss of connectivity) occur somewhere in the network, it can be very difficult to identify and localize the problem, leading to periods of lost scientific productivity across distributed resources which can last weeks to months and even to years, in some cases.

The particular network infrastructure that we will focus on in this paper is the one used by the scientific experiments at the LHC. The Worldwide LHC Computing Grid (WLCG) project[1] — a global collaboration of around 170 computing centers in more than 40 countries, is constructed on top of a conglomeration of local, regional, national and international networks, primarily research and education (R&E) networks but including some commercial segments. The mission of the WLCG project is to provide global computing resources to store, distribute and analyze peta- and exabyte-scale LHC data for the LHC experiments.

The LHC experiments at CERN currently transfer a few petabytes of data daily, in aggregate. While each data transfer is characterized by a single source and destination, the network path that is actually traversed between those endpoints can vary, based upon routing protocols and the status of the network components. The critical point to note here is that when there is a network problem, the first thing we need to know is "what path did the problem occur on?" Answering these questions identifies the relevant elements that could be the location for the problem being

observed. Fortunately, tracking of these dynamically changing paths or routes has become possible thanks to the massive and detailed network measurements being provided for the last several years by the OSG-LHC/WLCG network monitoring activity.[2] While the creation and operation of this network metric pipeline has given us the data we could use to find and localize network problems, in practice, it has given us an additional challenge in how best to identify, filter and visualize the massive amount of path information we are monitoring. This is the motivation for the creation of TRACER (TRACe route ExploRer)[3] and in the rest of this paper we will outline the technical details of TRACER and demonstrate how it is being used with OSG/WLCG data.

2. Background

In this section, we describe some of the relevant concepts and technologies used to define and measure network paths. The IP network path between a source and destination is determined by devices called routers, which operate at the IP layer (layer 3) in the OSI network stack. A network path can be described by the sequence of router IP addresses that packets traverse as they move from the source to the destination. Each router determines which next router to forward a network packet to (usually based only upon the destination network) using a routing table. Routing tables may be constructed automatically or manually. Manual routing tables comprises static routes while automatic produces dynamic routes. The routing table consists of at least three elements:

(1) network ID — the destination network;
(2) metric — a routing metric indicating the route priority;
(3) gateway — the address of next router to send the packet to.

WLCG routing tables are maintained semi-automatically, in most cases, using suitable software tools and frameworks. Even so, providing high performance at peak efficiency and availability becomes challenging for network operation staff. Even the slightest amount of packet loss on large bandwidth-delay paths drastically reduces the throughput achievable.[4] We note that it is possible to improve the behavior of TCP in the presence of packet loss but using different congestion control algorithms like TCP-BBR.[5] Maintaining networks in distributed computing systems would be impossible without monitoring and performance analysis tools. Network operation staff depend upon such tools to identify and fix network issues.

Toolkits, such as perfSONAR[6] (performance Service-Oriented Network monitoring ARchitecture), are used in WLCG for network testing and monitoring.[7] WLCG has deployed perfSONAR globally across its sites and has configured it to automatically monitor paths of interest, collecting and archiving network performance metrics in an efficient way. In 2020, we have more than 250 perfSONAR instances installed on 130 sites, constantly measuring and retrieving topology information, network delay, packet loss and bandwidth of network routes. The data retrieved by

perfSONAR tools are studied and analyzed in order to troubleshoot network performance problems. Machine learning methods have been applied for the prediction of performance issues based upon packet loss data measured by perfSONAR.[8] To better understand the data and identify issues in our networks, we rely on dashboards and visualization tools like Kibana and Grafana, both standard for data stored in Elasticsearch.[9] Kibana enables interested users to visualize data with histograms, line graphs, pie charts, sunbursts, tag clouds and more. Grafana is a powerful tool to query, visualize, and gain insights from data. Both of these tools are important components of WLCG network monitoring.[10]

Even with the monitoring, analysis and visualization tools, we have already described, we faced a challenge in analyzing and visualizing our measured network topologies. The problem is the complexity and dynamic nature of the path data. The data describing the network paths comes from either traceroute or tracepath tools and consists of the list of IP addresses of the routers along the path from source to destination, as well as additional information about the round-trip-time (rtt) to each router, the Autonomous System (AS) number which designates the network entity which manages each router, and, for tracepath, the maximum MTU supported to each router. Each router along the path constitutes a hop and the number of hops, even between a specific source and destination, can vary as conditions in the network change. Because our existing tools (Kibana, Grafana and Elasticsearch) were not well matched to the challenge of visualizing and understanding the network path data, we needed to develop TRACER. This paper describes TRACER, a 3D force-directed graph visualization for network routes. TRACER includes a web interface with filtering, data preparation functions and interactive visualization that allows users to monitor and explore network paths and how they change in time.

As TRACER was being developed, we engaged with the perfSONAR developer team, presenting early prototypes to get feedback on. The perfSONAR developers were enthusiastic about the new visualization capability for traceroute data and have had something similar on their future development list for many years. Given that the next release of perfSONAR (v4.4) will focus on visualization, we may have an opportunity to collaborate with them on improving integration between TRACE and perfSONAR.

3. Visualization Pipeline

Typical applications for visual analysis leverage specific visualization pipelines that include four main stages: sourcing, filtering, mapping and rendering. The sourcing stage provides the access to initial data sources. The filtering stage allows a user to select specific data for further analysis. The mapping stage is where filtered data items are mapped to geometric primitives (i.e. points, lines, spheres, cylinders) and their attributes, such as color, position or size. Mapping is the most critical stage for the efficiency and quality of the resulting visualization. The rendering stage is the process of transformating the geometric data to graphical images.

route visualization, and a right pane showing path details. In the following sections we detail the four main stages of our visualization pipeline.

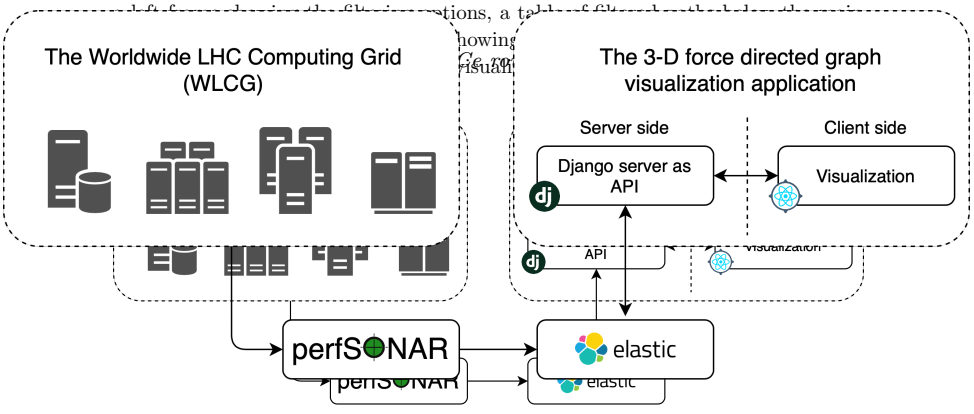

Fig. 1. The 3-D force directed graph visualization application architecture

Fig. 1. The 3D force directed graph visualization application architecture.

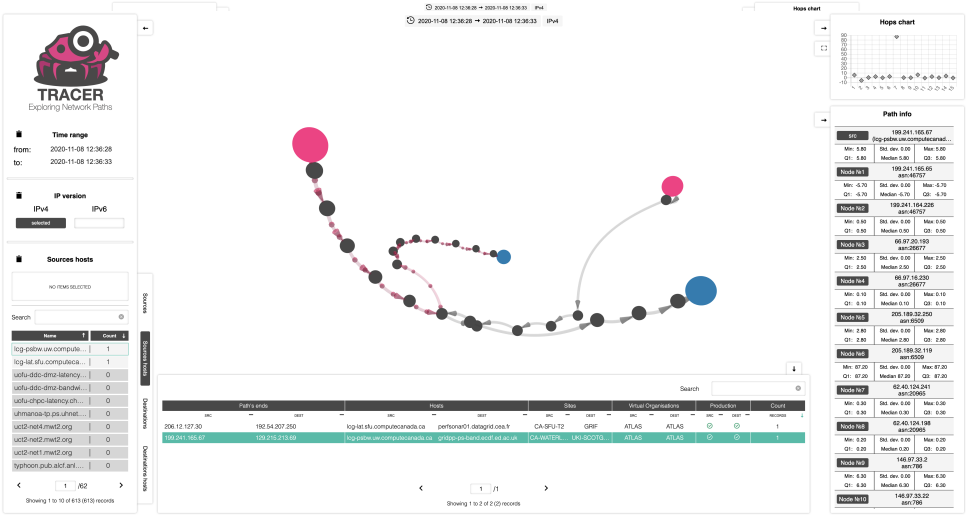

Fig. 2. TRACER web interface

The 3D force-directed graph visualization[11] includes both client and server components. For TRACER the server is based on Django which acts as an application programming interface (API) for the Elasticsearch data. The server accepts search queries and returns data retrieved from the Elasticsearch. The client provides a graphical user interface (GUI) to generate search queries and analyze and display the resulting data. The complex GUIs are built using the React JavaScript framework, which provides well-structured, easy to maintain source code. The application architecture is presented in Fig. 1. Figure 2 shows the main window of the TRACER web application. It consists of central frame for route visualization, a left frame showing the filtering options, a table of filtered paths below the main route visualization, and a right pane showing path details. In the following sections, we detail the four main stages of our visualization pipeline.

3.1. *Sourcing*

The data from WLCG perfSONAR servers is collected and transported to the Elasticsearch analytics cluster[a] at the University of Chicago. A sample of a network path record is shown in Figure 3.

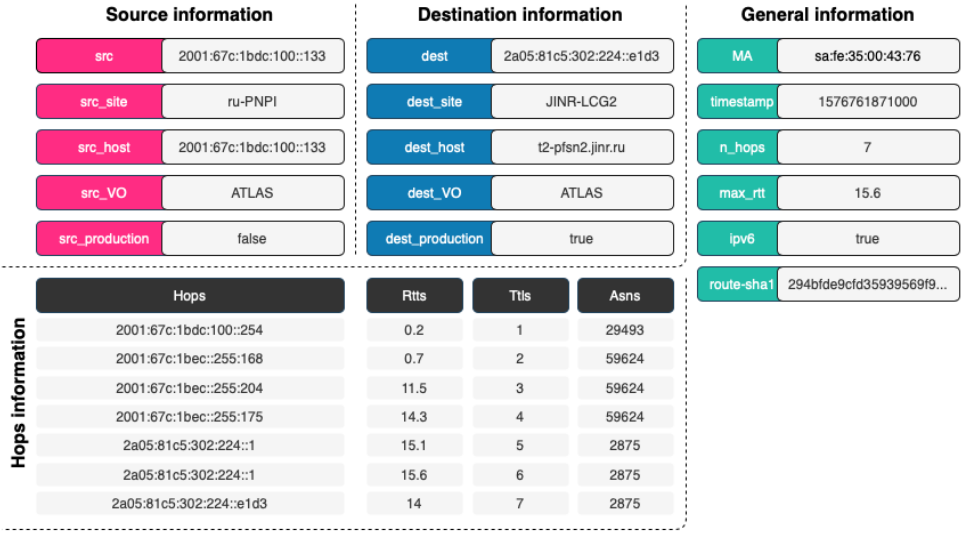

Fig. 3. The data record schema for the Network Route Visualization Tool.

Fig. 3. The data record schema for the Network Route Visualization Tool.

3.1. *Sourcing*

The data from WLCG perfSONAR servers is collected and transported to the Elasticsearch analytics cluster[a] at the University of Chicago. A sample of a network path record is shown in Fig. 3.

Each record describes network measurements of a source-destination network path at a particular point in time. When packets travel between a source and a destination, they pass through a number of intermediate network devices (routers). Each time a packet passes from one network segment to another a hop[12] occurs. Thus, network path record contains metadata about the endpoints and all hops between them. All routers in a path are identified by IPv4/IPv6 addresses. Each data record stored in our WLCG Elasticsearch is composed of the following three sets of parameters:

- **General Information**
  - *MA* — The Measurement Archiver (MA) that provided the original data.
  - *timestamp* - the time of the perfSONAR measurement.
  - *n_hops* - the number of hops on a route.
  - *max_rtt* - maximum value of the round trip time (rtt).
  - *ipv6* - IP protocol type (true if IPv6, false if IPv4).
  - *route_sha1* - cryptographic hash uniquely identifying a network path.

— *MA* — The Measurement Archiver that provided the original data.
— *timestamp* — the time of the perfSONAR measurement.
— *n_hops* — the number of hops on a route.
— *max_rtt* — maximum value of the round trip time (rtt).
— *ipv6* — IP protocol type (true if IPv6, false if IPv4).
— *route_sha1* — cryptographic hash uniquely identifying a network path.

- **Endpoints Parameters**
  - *src_site, dest_site* — source and destination WLCG site names.

[a]https://atlas-kibana.mwt2.org:5601/

— *src_host, dest_host* — source and destination host names.
— *ipv6* — version of IP protocol (ipv6 = True/False).
— *src_VO, dest_VO* — virtual organization of the src/dest WLCG owner.[b]
— *src_production, dest_production* — Boolean indicating source and destination production status (test versus production).

- **Router Parameters**
  — *round-trip time* (*rtts*) — the vector of round-trip-times (rtts) in ms to reach each router along the path.
  — *time to live* (*ttls*) — the vector of the values of the time-to-live field, decremented as a packet transits each router.
  — *autonomous system number* (*asns*) showing who operates each router.

The TRACER tool allows us to explore this network route visually. Each route measurement is stored in the Elasticsearch as a single document. There are couple of parameters that require further clarification:

- To identify unique paths through the network, we hash the vector of hops using the SHA1 hashing function to create route-sha1. This identifier allows us to easily track or aggregate on that specific path when measured by traceroute.
- The src_production or dest_production Booleans indicate whether a site is in production (true) or being tested (false) and is not related to the measurement.

3.2. *Filtering*

The filtering functions allow construction of search queries on the client-side. The filtering enables users to select time ranges, IP version (IPv4 or IPv6) and specific endpoint values: Sources (*src*), Source hosts (*src_host*), Destinations (*dest*), Destination hosts (*dest_host*). The filtering widgets are placed on the left side of the application window. The user query construction results in a query in JSON format compatible with the Elasticsearch engine. This query is passed from client to server where the Django application gets records from Elasticsearch and prepares the data for further graph building on the client.

3.3. *Mapping*

Five types of nodes are used to build force directed graph visualization. These node types are

- Source domain name — *src_host*,
- Source IP address — *src*,
- Router IP addresses — *hops*,
- Destination IP address — *dest*,
- Destination domain name — *dest_host*.

[b]https://wlcg.web.cern.ch/virtual-organisations

- Router IP addresses - *hops*
- Non-responding Router - (routers that didn't return their IP)
- Destination IP address - *dest*
- Destination domain name - *dest_host*

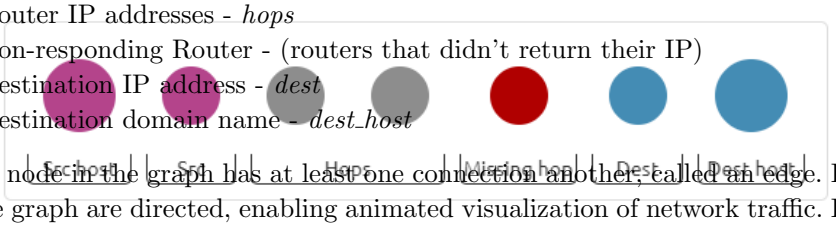

Fig. 4. (Color online) Nodes color scheme.

Every node in the graph has at least one connection another, called an edge. Edges on the graph are directed, enabling animated visualization of network traffic. Edges are set in compliance with paths, each having a weight assigned as the average rtts value of those measured. Edge weights determine the distance between nodes on the graph. Nodes of different types are highlighted with different colors: grey for routers, red for not responding routers, pink for source nodes and blue for destination nodes. To distinguish nodes of *src_host* from nodes of *src* the first ones have larger size, the same for *dest_host* and *dest* (Figure 4). There are cases when several paths are represented on the same graph, so it is possible for a node to be source and destination at the same time, then the node's color is set to purple. Nodes of *dest_host* and *dest* can also be highlighted in red when a path does not reach the actual destination (an incomplete path).

Fig. 4. Nodes color scheme.

3.4. *Rendering*

The rendering process on the client side exploits the WebGL technology through ThreeJS and D3 packages for JavaScript. The table of filtered paths (Figure 5) is displayed below the main visualization scene (Figure 6).

3.4. *Rendering*

Clicking a record in the table provides user with the additional information about the selected route in the right frame. It includes a violin chart (Fig. 7), explaining statistical distributions and probability density of rtts values of all hops, and a table describing hops and nodes of the network path (Fig. 8).

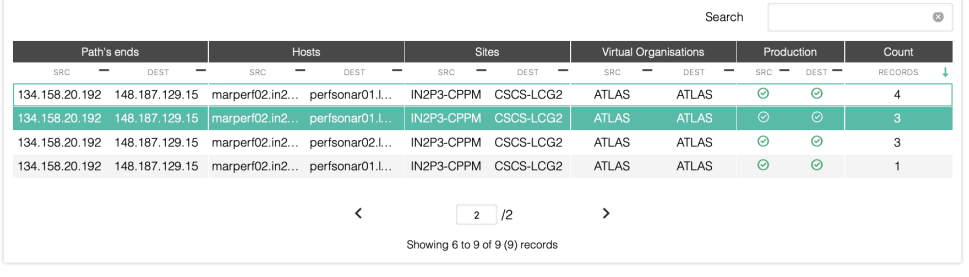

Fig. 5. Table with filtered paths.

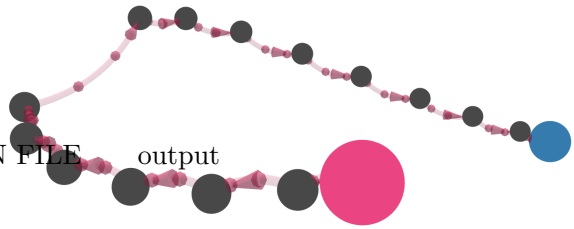

Fig. 6. Main visualization scene.

TRACER (TRACe route ExploRer): A Tool to Explore OSG/WLCG 9

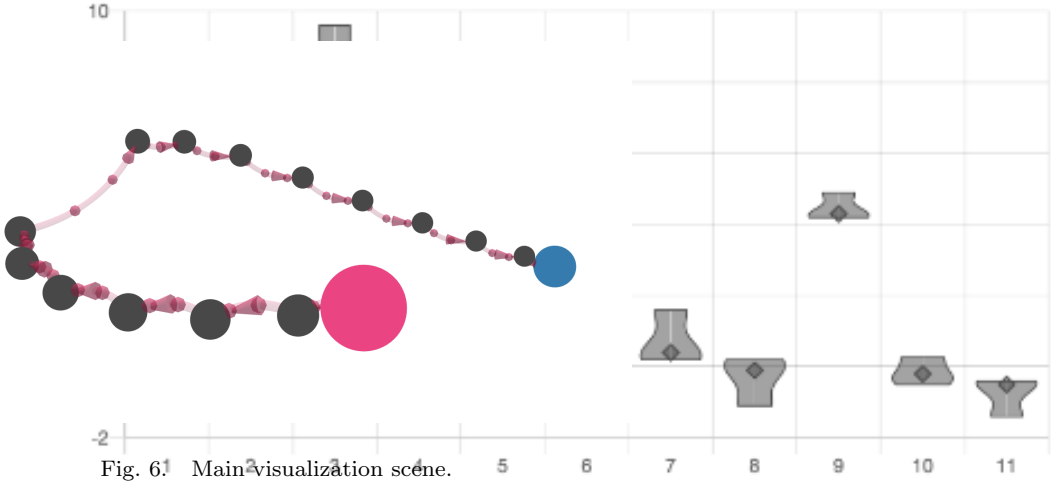

Fig. 6. Main visualization scene.

Fig. 7. Violin chart.

Fig. 8. Table with the detailed description of all nodes of network path

Path info

src	199.241.165.67 (lcg-psbw.uw.computecanad...	
Min: 5.80	Std. dev. 0.00	Max: 5.80
Q1: 5.80	Median 5.80	Q3: 5.80
Node №1	199.241.165.65 asn:46757	
Min: -5.70	Std. dev. 0.00	Max: -5.70
Q1: -5.70	Median -5.70	Q3: -5.70
Node №2	199.241.164.226 asn:46757	
Min: 0.50	Std. dev. 0.00	Max: 0.50
Q1: 0.50	Median 0.50	Q3: 0.50

Fig. 8. Table with the detailed description of all nodes of network path.

Clicking a record in the table provides user with the additional information about the selected route in the right frame. It includes a violin chart (Figure 7), statistical distributions and probability density of rtts values of all hops, describing hops and nodes of the network paths (Figure 8).

Hops chart

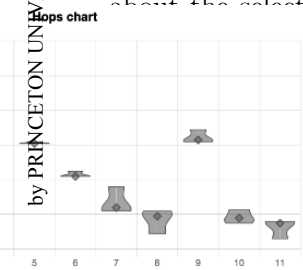

Fig. 7. Violin chart.

Users to l... network paths by providing visualiz... The application allows the user to ... segments, incomplete paths and dynamically v... perfSONAR deployment typically measures each source-destination pair between 3-4 times per hour. We see variations in the paths we find, with the majority of the time spent on long periods and others

Fig. 8. Table with the detailed description of all nodes of network path.

Clicking a record in the table provides user with the additional information about the selected route in the right frame. It includes a violin chart (Figure 7), statistical distributions and probability density of rtts values of all hops, and a table describing hops and nodes of the

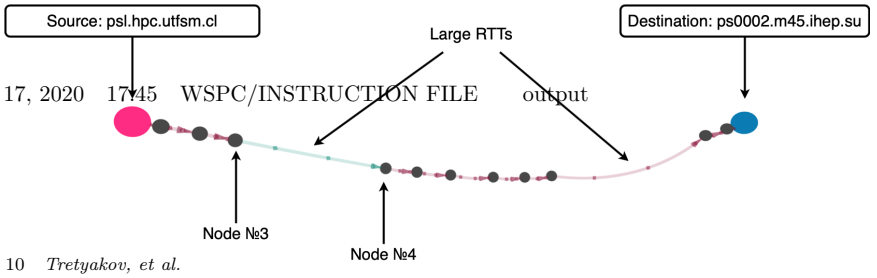

Fig. 9. Network path between psl.hpc.utfsm.cl and ps0002.m45.ihep.su sites.

fluctuating across many distinct paths. One cause of the fluctuation is the use of Equal-cost multi-path routing (ECMP)[13] which some networks use to distribute load across multiple paths to a destination.

Figure 9 presents a complete network path between two perfSONAR servers. This path can be considered as stable because it hasn't been changed for quite some time and does not have missing hops. One interesting feature to note in this path is that the hop between 200.0.204.153 (Node 3) and 200.0.204.58 (Node 4) routers is 113.60 milliseconds. These two routers are both part of the RedCLARA network, with the first in Chile and the second in Mexico. The second large RTTS is due to a hop across the Atlantic Ocean.

Figure 10 illustrates the path with self-looped segment, which should not happen during normal routing. We analyze all paths to determine if a router (IP) appears more than once in the path and if it is found we mark the path as looping. The router with IP 205.189.32.54 is encountered twice on the route. TRACER allows us

fluctuating across many distinct paths. One cause of the fluctuation is the use of Equal-cost multi-path routing (ECMP)[13] which some networks use to distribute load across multiple paths to a destination.

4. Results

Figure 9 presents a complete network path between two perfSONAR servers. This path can be considered as stable because it hasn't been changed for quite some time and does not have missing hops. One interesting feature to note in this path is that the hop between 200.0.204.153 (Node 3) and 200.0.204.58 (Node 4) routers is 113.60 milliseconds. These two routers are both part of the RedCLARA network, with the first in Chile and the second in Mexico. The second large RTTS is due to a hop across the Atlantic Ocean.

TRACER helps users to better understand the topology of network paths by providing filtering and visualization of network topology details. The application allows the i complete path l loyment typi pe. We see vari d others fluct ıe use of Equ istribute load

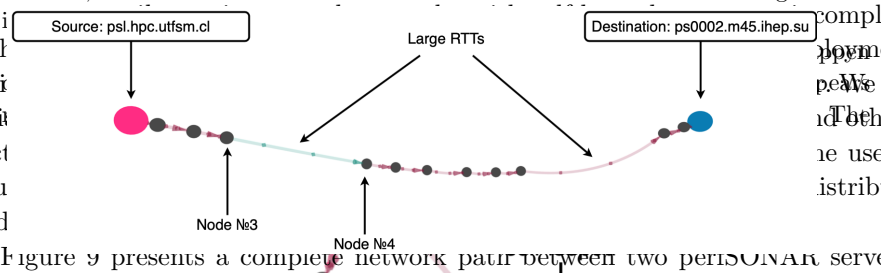

Fig. 9. Network path between psl.hpc.utfsm.cl and ps0002.m45.ihep.su sites.

Figure 9 presents a complete network path between two perfSONAR servers. This path can be considered as stable because it has not been changed for quite some time and does not have missing hops. One interesting feature to note in this path is that the hop between 200.0.204.153 (Node 3) and 200.0.204.58 (Node 4) routers is 113.60 milliseconds. These two routers are both part of the RedCLARA network, with the first in Chile and the second in Mexico. The second large RTTS is due to a hop across the Atlantic Ocean.

Figure 10 illustrates the path with self-looped segment, which should not happen during normal routing. We analyze all paths to determine if a router (IP) appears more than once in the path and if it is found we mark the path as looping. The

Figure 10 illustrates the path with self-looped segment, which should not happen during normal routing. We analyze all paths to determine if a router (IP) appears more than once in the path and if it is found we mark the path as looping. The

ı ws us

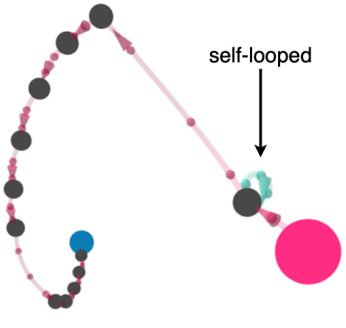

Fig. 10. Network path between CA-SFU-T2 and INFN-MILANO-ATLASC sites.

Fig. 10. Network path between CA-SFU-T2 and INFN-MILANO-ATLASC sites.

by routing misconfigurations or when internet routes are in the process of changing or updating.

Figure 11 illustrates case when certain routers, due to their configuration or a technical malfunction, do not respond to a request from perfSONAR that measures the network path. Loss of network path data is observed form 9th hop to the desti-

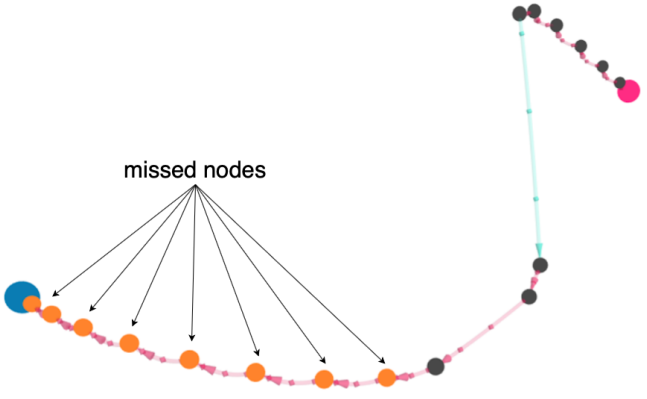

Fig. 11. Network path between RAL-LCG2 and NCP-LCG2 with missed nodes.

Fig. 11. Network path between RAL-LCG2 and NCP-LCG2 with missed nodes.

router with IP 205.189.32.54 is encountered twice on the route, TRACER allows us nation, because relevant routers do not respond with repeated measurements. But to spot such self-looped segments visually easily. Such looping paths can be caused the path itself reaches the destination, thus it can be assumed that these losses by routing misconfigurations or when internet routes are in the process of changing occur due to the security settings of the respective routers. or updating.

Figure 11 illustrates the case when certain routers, due to their configuration or a technical malfunction, do not respond to a request from perfSONAR that measures

5. Future Work

The network path. Loss of network path data is observed from 9th hop to the This first release of TRACER provides a functional service for exploring the com- destination, because relevant routers do not respond with repeated measurements. plex, dynamic topologies measured by the global deployment of perfSONAR in- But the path itself reaches the destination, thus it can be assumed that these losses, stances in hundreds of locations worldwide. Its primary use-case is to support users, occur due to the security settings of the respective routers.

5. Future Work

site-admins and network administrators in better understanding the complex net- work topology involved in our WLCG infrastructure. However, as with almost any service or tool, we already see areas needing improvement as well as opportunities for new features and capabilities. The following lists the primary areas we intend to work on in the future:

This first release of TRACER provides a functional service for exploring the com- plex, dynamic topologies measured by the global deployment of perfSONAR in- stances in hundreds of locations worldwide. Its primary use-case is to support users, site-admins and network administrators in better understanding the complex net- work topology involved in our WLCG infrastructure. However, as with almost any service or tool, we already see areas needing improvement as well as opportunities for new features and capabilities. The following lists the primary areas we intend to work on in the future:

- TRACER can be very slow or will not even respond if the user selects a time period of more than a few minutes without selecting additional filtering.
- TRACER currently displays only data about network traces, but a very interesting extension would be to allow it to gather and display associated metrics along a set of identified paths, e.g. latency, throughput and packet-loss.
- TRACER could be used to explore how paths change in time, and updating the user interface to support this would be a valuable extension.

- TRACER can be very slow or will not even respond if the user selects a time period of more than a few minutes without selecting additional filtering.
- TRACER currently displays only data about network traces, but a very interesting extension would be to allow it to gather and display associated metrics along a set of identified paths, e.g. latency, throughput and packet-loss.
- TRACER could be used to explore how paths change in time, and updating the user interface to support this would be a valuable extension.

- Visualization of the intersections of traces of multiple source-destination pairs where some issues were detected allows us to visually understand path correlations. Additionally, TRACER may be used for the visualization of all network paths that use one particular router.
- Identifying asymmetric routes which can be associated with network configuration problems. The challenge here is in two parts: (1) we need to identify all the IP addresses routers utilize, since a given router may report a different IP depending upon how it is traversed and (2) we need to improve the visualization to better support this use case (similar to the previous item above) so we can clearly visualize source-to-destination and destination-to-source to identify asymmetries.
- TRACER visualizes routes in terms of router hops but we also have data about the AS number (network owner) and this can allow us to provide a much more condensed view of the path, highlighting the major network entities along the path.

With the updated, refactored code in this first release, we believe that much of the above should be quickly achievable and expect at least few of the items above will be addressed in our next release.

6. Conclusion

We have described the motivation, design and implementation of TRACER, a tool for visual exploration of network topology measured by the global deployment of OSG and WLCG perfSONAR toolkit instances. TRACER is now integrated in CERN computing infrastructure and can be used for the near real-time network monitoring and retrospective analysis of network faults and malfunctions. An additional use-case is interactive visualization to support the development of mathematical and statistical methods for the detection of network failures.

Acknowledgments

The development of the visualization platform is financially supported by the Russian Science Foundation award #18-71-10003 (research conducted in Lomonosov Moscow State University). We also gratefully acknowledge the Russian Science Foundation award #19-71-30008 (for research related to the Russian scientific data lake prototype, this research is conducted in Plekhanov University, Moscow) and the US National Science Foundation which supported data collecting, storing and processing part of this work through NSF grants #1836650 and #1827116.

References

1. Y. Bird, P. Buncic, F. Carminati, M. Cattaneo, P. Clarke, I. Fisk, M. Girone, J. Harvey,B. Kersevan, P. Mato, R. Mount and B. Panzer-Steindel, Update of the Computing Models of the WLCG and the LHC Experiments, Tech. Rep. CERN-LHCC-2014-014.LCG-TDR-002, CERN (Apr 2014).

2. M. Babik, S. McKee, B. P. Bockelman, E. M. F. Hernandez, E. Martelli, I. Vukotic, D. Weitzel and M. Zvada, Improving WLCG networks through monitoring and analytics, in *EPJ Web Conf.*, Sofia, Bulgaria (2019), p. 08006.
3. E. Tretyakov and I. Vukotic, sand-ci/trace-explorer-tracer: First release, doi:10.5281/zenodo.3903716 (2020).
4. T. V. Lakshman and U. Madhow, *IEEE/ACM Trans. Network.* **5**, 336 (1997).
5. M. Hock, R. Bless and M. Zitterbart, Experimental evaluation of bbr congestion control, in *2017 IEEE 25th Int. Conf. on Network Protocols* (*ICNP*) (2017), pp. 1–10.
6. B. Tierney, J. Boote, A. B. E. Boyd, M. Grigoriev, J. Metzger, M. Swany, M. Zekauskas and J. Zurawski, Instantiating a global network measurement framework, in *SOSP Workshop on Real Overlays and Distributed Systems* (*ROADS'09*), Big Sky, MT, USA, 2009.
7. S. Campana *et al.*, *J. Phys.*: *Conf. Ser.* **513**, 062008 (2014), doi:10.1088/1742-6596/513/6/062008.
8. M. Babik and H. Borras, Simulating network throughput by correlating perfSONAR measurements with link utilisation, Tech. Rep. CERN-IT-Note-2017-001, CERN (2017).
9. Elastic-Developers, Elastic stack, https://www.elastic.co/guide/index.html, (2018).
10. S. McKee, Prototype OSG/WLCG kibana network dashboards, https://atlas-kibana.mwt2.org/s/networking/app/kibana#/dashboard/07a03a80-beda-11e9-96c8-d543436ab024?_g=() (2020).
11. Wikipedia contributors, Force-directed_graph_drawing — Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Force-directed_graph_drawing (2020).
12. Wikipedia contributors, Hop (networking) — Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Hop_(networking) (2020).
13. Wikipedia contributors, Hop (networking) — Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Equal-cost_multi-path_routing (2020).