Revealing the Evolution of a Cloud Provider Through its Network Weather Map

Maxime Piraux
UCLouvain
Belgium
maxime.piraux@uclouvain.be

Louis Navarre
UCLouvain
Belgium
louis.navarre@uclouvain.be

Nicolas Rybowski
UCLouvain
Belgium
nicolas.rybowski@uclouvain.be

Olivier Bonaventure
UCLouvain
Belgium
olivier.bonaventure@uclouvain.be

Benoit Donnet
Université de Liège
Belgium
benoit.donnet@uliege.be

ABSTRACT

Researchers often face the lack of data on large operational networks to understand how they are used, how they behave, and sometimes how they fail. This data is crucial to drive the evolution of Internet protocols and develop techniques such as traffic engineering, DDoS detection and mitigation. Companies that have access to measurements from operational networks and services leverage this data to improve the availability, speed, and resilience of their Internet services. Unfortunately, the availability of large datasets, especially collected regularly over a long period of time, is a daunting task that remains scarce in the literature.

We tackle this problem by releasing a dataset collected over roughly two years of observations of a major cloud company (OVH). Our dataset, called *OVH Weather dataset*, represents the evolution of more than 180 routers, 1,100 internal links, 500 external links, and their load percentages in the backbone network over time. Our dataset has a high density with snapshots taken every five minutes, totaling more than 500,000 files. In this paper, we also illustrate how our dataset could be used to study the backbone networks evolution. Finally, our dataset opens several exciting research questions that we make available to the research community.

CCS CONCEPTS

• **Networks** → **Network measurement**; *Data center networks.*

KEYWORDS

Cloud infrastructure, Network topology, Dataset

ACM Reference Format:
Maxime Piraux, Louis Navarre, Nicolas Rybowski, Olivier Bonaventure, and Benoit Donnet. 2022. Revealing the Evolution of a Cloud Provider Through its Network Weather Map. In *ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3517745.3561462

IMC '22, October 25–27, 2022, Nice, France

ACM ISBN 978-1-4503-9259-4/22/10…$15.00
https://doi.org/10.1145/3517745.3561462

1 INTRODUCTION

Network researchers often design new tools, techniques, and protocols that are eventually deployed in real networks. To evaluate their proposals, researchers need to use data such as network topology, traffic distribution, and router configuration. While such data is collected and archived by most network operators, it is rarely made available to researchers. Fortunately, over the years, network operators and the research community have collected and released public datasets.

These datasets have enabled significant advances in the field and some have been widely used. Using packet traces, researchers have shown that Internet traffic could not be modeled using Poisson processes [21, 31]. Some researchers continue to collect packet traces that are used by many different studies [9, 42]. Various network topology maps [20, 28, 38] can be used by researchers when testing routing protocols. Traffic engineering techniques can be tested using synthetic [16] and real traffic matrices [46]. Measurement platforms such as Ripe Atlas [40], PlanetLab [39], or M-Lab [14, 17] enable researchers to perform active measurements from many different locations. Finally, regular `traceroute` campaigns [10, 40] provide information about interface- and router-level topology [12].

Despite these datasets, researchers rarely have access to data on operational networks. Abilene/Internet2 was an exception with the Abilene observatory, but this is a research backbone differing from commercial networks. Furthermore, some existing datasets have limitations. For example, while `traceroute`-based tools can extract the topology of a network [23], the inferred topology does not perfectly match the real one [41]. Given the limited number of publicly available traffic matrices [46], most traffic engineering studies rely on synthetic traffic matrices [16, 45].

Fortunately, some network operators expose information about their network by providing BGP feeds on Routeviews or Ripe RIS, or by installing *Looking Glass* servers. Some also expose their full network topology and the load of their links using software such as `PHP Weathermap` [1]. This open-source software uses SNMP to collect link load information, typically every five minutes, and exposes through an interactive website a network map. However, the raw data is not released and users only have access to the image produced by `PHP Weathermap`. In this work, we collect, analyze and release SVG images produced by the OVH Network Weathermap. OVH is a French cloud provider with more than 300,000 servers

spread in 32 datacenters and a world-wide network of more than 180 routers with a total egress capacity of more than 20 Tbps.

This paper makes three contributions. First, we show that the OVH Network Weathermap can be used to automatically extract the topology and the link load levels of a large network from its SVG images. Second, we release our dataset, called the *OVH Weather dataset*, consisting of the original data collected from the OVH Network Weathermap over almost two years with snapshots taken every five minutes, the associated processed files, processing scripts, and wrapper scripts. This dataset is available at **`https://weathermapdataset.info.ucl.ac.be/`**. Finally, we briefly analyze the dataset and its main characteristics to illustrate how the data can be leveraged for future research.

The remainder of this paper is organized as follows. Section 3 positions this work with respect to the state of the art. Section 4 describes how weather maps were collected and processed into a format enabling their analysis. Section 5 illustrates how our dataset can be used to study the evolution of CDN backbone networks. Finally, Section 6 concludes this paper by discussing how researchers could leverage the OVH Weather dataset for their own research.

2 ETHICS

This work does not raise any ethical issues. First, the data is collected from a public web site offering several web pages without any restriction or text notice. We collect and archive the data in the purpose of scientific research and education. We believe making this data available in our dataset does not violate the rights of the copyright owner. Second, the data does not identify users of the network. It lists the links between routers and physical peerings, along with their relative level of usage expressed in percentage. No IP addresses nor flow-level information are available. When processing the data, we do not make any effort in identifying users of the network. We believe the level of aggregation of the data as disclosed by the copyright owner is already sufficient to prevent any ethical issue. We contacted OVH after this work and presented our results to them.

3 RELATED WORK

During the past twenty years, efforts have been made by the research community to collect and share datasets. First, Internet topology data has attracted a lot of interest [12]. At the IP interface level, data can be collected through `traceroute` measurements [7, 10, 26, 35] or through multicast probing [26] towards the Internet but also towards specific ISPs [38]. Reverse DNS queries can also reveal information on the growth and change in ISPs networks [15]. Sometimes, ISP maps can be directly collected from the operators' published information [20]. IP interfaces can then be aggregated at the router level through the alias resolution process [18, 19, 24, 43]. Autonomous System (AS) level data can also be gathered through publicly available BGP information [32, 44, 47].

Those topologies have been used for, e.g., modeling the Internet [12, 30], understanding peering [22], conducting relevant traffic engineering (e.g., the REPETITA framework [16]) or as ground truth for alias resolution (e.g., Sherry et al. [36]). Rocketfuel topologies are now outdated [25] and the Zoo dataset [20] is more than 10 years old.

Second, traffic traces have also been shared such as Internet traffic traces (e.g., CAIDA's Telescope [42], or MAWI data [9]), traffic matrices (e.g., GEANT [46] or Abilene [49]), operator traffic traces (e.g., SDNLib [28]), datacenter traffic (e.g., [6]), or even mobile devices traffic (combining physical, network, transport-layer and geographical data [37]). Operator traffic traces may be used, for instance, to compare design models and algorithms. Internet-wide data is mainly used for security analysis, such as studies on DDoS attacks [27].

More recently, cloud infrastructure data has been shared. For instance, the Alibaba workload [8], AWS Reserved Instance Marketplace (RIM) data [5], or Facebook traffic patterns [34] are datasets available from major cloud companies. Finally, datasets have been published for evaluating the performance of virtualized network functions (VNFs) [33].

Most of those datasets provide a single snapshot as they consist of data that is collected once and shared with the community. This is different from the data presented in this paper as we provide a longitudinal dataset, with snapshots taken every five minutes during a period spanning almost two years.

4 DATASET COLLECTION AND PROCESSING

The OVH Network Weathermap. The OVH Network Weathermap [3] is a website made available by OVH to the public for more than fifteen years. Archived pages show early versions of the website in which several graphs representing the company network status are available. These pages reported the OVH network internal routers and links, the physical peering links, and the links loads in percentage. By navigating through the archived pages, one can glance at the evolution of this company network at a very coarse level of detail in time.

Today, the OVH Network Weathermap contains additional map images with a greater number of routers and links as its network infrastructure has greatly expanded. We observed increments in time between two map updates and posit that they are updated every five minutes. When a map is updated, the most recent snapshot is replaced with the updated one. The website only keeps past snapshots of the day at a granularity of one hour. We discern two kinds of maps: (i) maps of the core network infrastructure within a given OVH datacenter and which do not contain peering links; (ii) maps of the backbone network. There exists four of these latter maps: Europe, which has historically been the largest network infrastructure, World, North America, and Asia Pacific. The World weather map only contains intercontinental links between routers of the other maps. Combining the different maps together yields a global overview of the network. The OVH Weather dataset focuses on the backbone weather maps.

Figure 1 represents a small part of the Europe map to illustrate the elements composing a weather map. An OVH router is represented by a white box and a lower case name, such as `fra-fr5-pb6-nc5`. A physical peering is represented by a white box and an upper case name, such as `ARELION` on the right-hand corner. Two meeting arrows represent a bidirectional link. The `fra-fr5-pb6-nc5` router has several links towards different physical peerings. Each arrow also reports the link load in its direction, explicitly with a percentage and implicitly through its color. In our example, the two horizontal

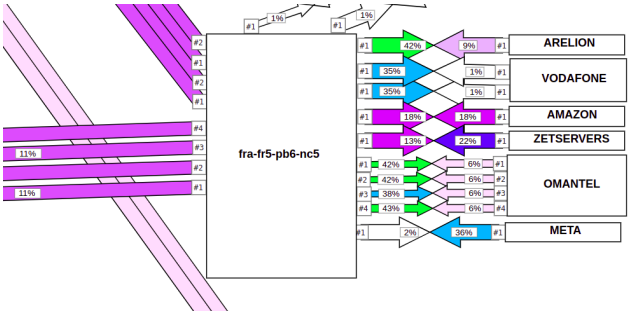


Figure 1: Part of the Europe map showing one OVH router, several peerings, associated network links, and links loads.

Network Map	OVH routers	Internal links	External links
Europe	113	744	265
World	16	76	0
North America	60	407	214
Asia Pacific	23	96	39
Total	181	1,186	518

Table 1: Summary of the routers, internal links, and external links on Sept 12, 2022.

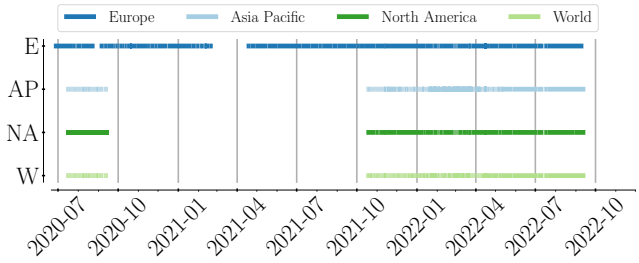


Figure 2: Collected data time frame by network weather map.

arrows in the top right corner represent a link between the OVH router and the ARELION peering which is used at 42 % (resp. 9 %) of its egress (resp. ingress) capacity from the OVH perspective. A disabled link is represented with a load level of 0 %. Each link is also labeled in both directions (for instance, the aforementioned link has the label #1 in both directions). As depicted in Figure 1, several parallel links can connect two routers (e.g., between fra-fr5pb6-nc5 and OMANTEL). Some parallel links, such as the ones connecting the VODAFONE peering, can have non-unique labels. OVH routers can also be connected together, as illustrated by the arrows pointing west of the fra-fr5-pb6-nc5 router, eventually reaching other routers outside the scope of our example.

Table 1 illustrates the OVH network size with the number of elements in the four weather maps on the 12th of September 2022. The Europe map represents the largest network infrastructure. The World map does not contain peering links as it connects intercontinental routers. The North America map is the second largest map, with approximately half the size of the Europe map. The Asia Pacific map is the smallest map. The total takes into account routers appearing simultaneously in several maps.

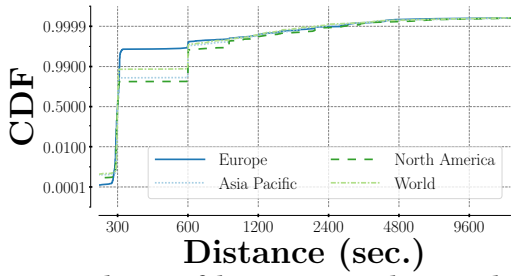


Figure 3: Distribution of distance in time between data files by network weather map.

Collecting the dataset. We started collecting the maps of the OVH Network Weathermap in July 2020 by downloading them every five minutes. Figure 2 illustrates the time frame spanned by the data we collected. For each map, segments cover the time for which snapshots are available every five minutes. The Europe map is the most available map in our dataset. The World, North America, and Asia Pacific maps were collected between July and September 2020 and after October 2021. When looking closely at Figure 2, one can observe discontinuities. To better understand the gaps that exist within our dataset, Figure 3 reports the distribution of time intervals between consecutive snapshots of the maps. For the Europe map, more than 99.8 % of the snapshots are available at the highest resolution of five minutes. For the three other maps, the resolution can be coarser less than 10 % of the time but in a very large amount of cases the gap is not larger than ten minutes, corresponding to one missing snapshot. In May 2022, we identified and fixed an operational issue impacting the data collection. As less short gaps appear in Figure 2 past this point, the fix improved our data collection.

Processing the dataset. On the OVH Network Weathermap, maps are available as SVG files, i.e., vector images based on XML, which can be processed by scripts to extract information. However, these files pose a challenge as the SVG tags representing elements of the map, i.e., routers, links, labels, and usage percentages, are not all hierarchically organized. Instead, the SVG file lists the elements of the map in a flat manner with coordinates positioning them in the 2D image space. Thus, from a tag representing a link label, one has to find back the link it belongs to. Then, the link has to be associated to the routers it connects.

To achieve this, we developed a Python script to extract the routers, their links, labels, and links loads from an SVG file and to output this information in a YAML file. The script leverages the geometric objects, their order in the file and their placements in the 2D image space to find back their relationships.

Our script operates in two steps. First, it extracts the relevant information from the SVG file and finds out some of their relationships based on the SVG tags orderings. Algorithm 1 describes this process. It iterates over SVG tags and inspects their *class* and *tag* type to determine whether they represent a router, a part of a link, or a link label. For a router or a physical peering, the coordinates of its white box and its name are extracted (Lines 7–8). Two successive polygon SVG tags represent the two arrows of a bidirectional link. They are parsed sequentially and their coordinates are extracted (Lines 9–13). The two load levels follow the two arrows in an SVG file and are parsed sequentially (Lines 14–15). At this stage, the link

Algorithm 1 SVG parsing to objects

```
1:  links ← ()
2:  routers ← ()
3:  labels ← ()
4:  link ← nil                          ▷ Temporary variable when parsing a link
5:  label ← nil                         ▷ Temporary variable when parsing a label
6:  for each elem ∈ svg do
7:      if elem.class starts with object then                    ▷ Router
8:          routers.add(elem)
9:      else if elem.tag is polygon then                         ▷ Link arrow
10:         if link is nil then                                  ▷ First arrow of the link
11:             link ← elem.coords
12:         else                                                 ▷ Second arrow of the link
13:             link.extend(elem.coords)
14:     else if elem.class is labellink then                     ▷ Load (%)
15:         link.loads.add(elem.text)
16:         if |link.loads| = 2 then
17:             links.add(link)                                  ▷ Link is complete
18:             link ← nil
19:     else if elem.class is node then                          ▷ Label
20:         if elem.tag is rect then                             ▷ Label box
21:             label ← elem
22:         else if elem.tag is text then                        ▷ Label text
23:             label.text ← elem.text
24:             labels.add(label)                                ▷ Label is complete
```

is added to the extracted information (Lines 16–18). Link labels are also parsed sequentially in two steps, with first their white boxes (Lines 20–21) and then their text (Lines 22–23). At completion, Algorithm 1 outputs an extracted list of routers, a list of links, and of link labels.

Algorithm 2 Object attribution

```
1: for each link ∈ links do
2:     line_link ← line(link)
3:     r_∩link ← (r|r.coords ∩ line_link, r ∈ routers)
4:     l_∩link ← (l|l.coords ∩ line_link, l ∈ labels)
5:     for each end ∈ link.ends do
6:         Sort lists by distance to this end of link
7:         end.router ← first(r_∩link)                          ▷ Connect link end
8:         end.label ← first(l_∩link)                           ▷ Attribute its label
9:         labels ← labels \ (end.label)                        ▷ Remove the label
```

Algorithm 2 then finds the relationship between elements of the three lists by leveraging the shape and placement of their SVG tags in the 2D image space. For each link, it first computes the straight line in the 2D space represented by a link with the middle coordinates of the basis of the two arrows of the link (Line 2). Then, it computes the lists of routers (Line 3) and link labels (Line 4) intersecting this line. For each of the two ends of a link, the lists are sorted in increasing distance to the end (Line 6). The end is connected to its closest router (Line 7) and is attributed the closest link label (Line 8). At completion, Algorithm 2 has associated all link labels, links, and routers. As a final step, these associations are formatted in a YAML file.

Network Map	SVGs		YAMLs	
	# Files	Size (GiB)	# Files	Size (GiB)
Europe	214,426	161.39	214,340	20.16
World	111,459	6.22	111,431	0.83
North America	107,088	50.64	107,024	6.23
Asia Pacific	109,076	9.67	109,024	1.24
Total	542,049	227.93	541,819	28.46

Table 2: Summary of the collected and processed files in our dataset on Sept 12, 2022.

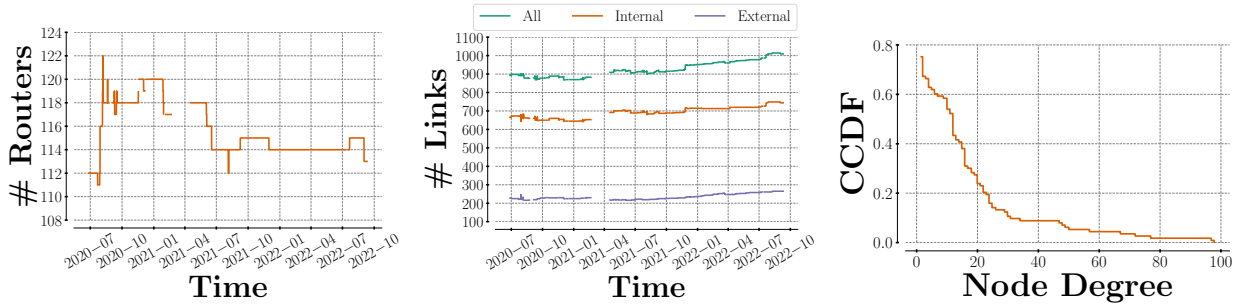
Parsing sanity checks. We added tests when extracting the topology information from the SVG files. We first ensure that each link load lies within $[0, 100]$. We also verify that each link is constructed from two arrows. The most critical part of processing is the object attribution. For instance, a router may be attributed to several links but a given label belongs to a unique link end. To that end, we first assert that the distance between the link end and its label is below a defined threshold (i.e., a few pixels). Second, when we attribute the label to its link end, we remove the label from the labels set (Line 9 in Algorithm 2) to ensure that labels get assigned to a link only once. Upon completion, we ensure that each router is attributed at least one link. The scripts report an error when a link is not connected to two (distinct) routers.

The OVH Weather dataset. Table 2 summarizes the data collected and processed up to the 12th of September 2022. For each map, the number of files and their total size for each file type is provided. Almost all the SVG files were processed by our script to produce YAML files, leaving less than a hundred files per map unprocessed. There are several causes for being unable to process them. First, we observed some SVG files to be invalid, e.g., with malformed attribute values. The reasons for these errors are not known. Second, some SVG files are lacking elements, such as OVH routers, resulting in a failure to find intersections for a given link. We may posit that these files are produced when the OVH Network Weathermap lacks status information for some part of the company infrastructure due to an error unrelated to the network status.

5 DATASET ANALYSIS

In this section, we analyze the Europe map in our dataset to illustrate the diversity of the behaviors it contains and how it can be used to study the evolution of backbone networks. We first report the evolution of the network infrastructure. Then, we study its links loads to show how the network is provisioned and used, as well as the effectiveness of traffic engineering techniques used by OVH to spread the traffic load over parallel links. Finally, we study the effect of a link upgrade in the backbone network. Except when stated otherwise, all Figures are produced with all the data available for the Europe map.

Network architecture evolution. Figure 4 illustrates the network infrastructure of the Europe map. First, Figure 4a reports the evolution of the number of OVH routers. Over the data collected, the OVH network has marginally grown but several changes can be observed. Over the course of August 2020 to September 2020, ten routers were added to the Europe map. Four routers were removed shortly after this increase. Four more routers were removed from the map in June 2021. A short decrease can be observed in August



(a) Evolution of the number of OVH routers. **(b) Evolution of the number of links.** **(c) Node degree distribution of OVH routers on September 12th, 2022.**

Figure 4: Network infrastructure of the Europe map.

2021. These events of *increase then decrease* could be attributed to upgrades of the network infrastructure in a *make-before-break* manner, while *decrease then increase* could indicate forced maintenance events and failures forcing OVH to temporarily remove routers from its infrastructure.

Second, Figure 4b presents the evolution of the number of links. It discerns internal from external links to better highlight changes in the OVH backbone network from changes with its peerings. We can observe that the number of external links gradually increased over time. As a cloud provider, OVH relies on peerings to offer its services to the Internet. Increasing the number of links to other networks improves its services availability and resilience. The number of internal links in the OVH network also increased over time, but the increases mostly occurred by steps. For instance, in November 2021, an important event of increase can be observed. This difference could be explained by the way link increases are planned, e.g., new peering links could be added separately while core network upgrades happen in a more coordinate manner. Future work could use router names to identify the spread of these variations in the network, e.g., to find whether some parts of the network are growing faster than others.

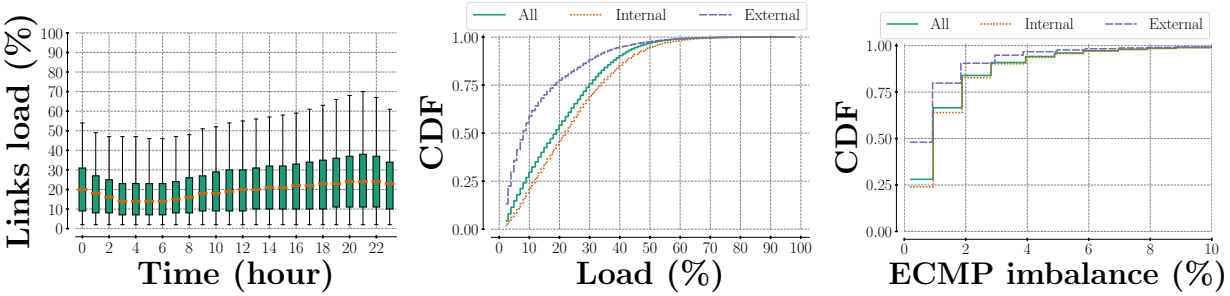
Finally, Figure 4c presents the complementary cumulative distribution function of the node degree of the OVH routers computed as the number of links connected to a given router, including all parallels links. More than 20 % of the OVH routers on the Europe map are connected with a single link. As the map does not contain all connections of a given router outside the OVH backbone network, for instance to other routers in a particular datacenter, these routers appear as isolated nodes on the map. We can also observe that the OVH backbone network has an important number of routers connected with several links, e.g., more than 20 % of the OVH routers have more than 20 links. The network topology thus presents path diversity among the core routers, which can be leveraged for instance by traffic flowing between datacenters.

Links loads. Figure 5 represents the links loads in the Europe map. First, we study the evolution of the distribution of links loads over hours of the day in Figure 5a. The links loads are extracted from all files of the Europe map and grouped into hours of day. The Figure 5a reports the median load value in orange, the 25th and 75th percentile in green and whiskers indicate the 1st and 99th percentile. We can make two observations. First, the median load value follows a sinusoidal form over the day, reaching its lowest point between 2 and 4 a.m. and its highest point between 7 and 9 p.m., following a typical day cycle. Second, when the network is more loaded, the variance of the distribution of loads increases.

Then, Figure 5b represents the distribution of links loads from the data on the Europe map. We observe that 75 % of the loads are below 33 % and very few loads exceed 60 %. It shows that the OVH network has an excess capacity, which can be used to accommodate maintenance without impacting the volume of traffic transiting through the OVH network, and that congestion inside the network happens occasionally. Furthermore, links loads of external links are on average lower than OVH internal links loads. This could be explained by the excess capacity being more important for external rather than internal links. While we observed that external links are constantly added to the Europe map, we posit that the time to decide and implement a link upgrade is higher with external than internal links, requiring OVH to provision capacity more significantly. Further, as OVH is a cloud provider, it cannot directly control the traffic originating from its network, requiring again careful planning of its network capacity.

Finally, the Europe map has an important number of parallel links. On the 12th of September 2022, OVH routers had in average 6.58 parallel links. When several network paths are available between two hosts, traffic engineering techniques such as ECMP are used to spread the traffic between them. We can compute the load imbalance for each group of parallel links to evaluate the effectiveness of these techniques in the OVH network. We compute the load imbalance as the difference between the maximum and the minimum load for each directed set of parallel links. We ignore links with 0 % load as they are unused in the network. We also discount links with 1 % load as we cannot differentiate a low traffic load value from control traffic only. Then, we remove sets with only one remaining link.

Figure 5c reports the distribution of the computed imbalances in the Europe map over the entire observation period for internal and external links. We assume that all parallel links between two routers have the same capacity. We can note that more than 60 % of the imbalance values are lower or equal to 1 %. This indicates that the traffic engineering techniques applied are effective in an important number of cases. Furthermore, external links have in average a lower imbalance than internal links with more than 90 %



(a) Links load per hours. (b) Distribution of links loads. (c) Distribution of ECMP load imbalance.

Figure 5: Links loads in the Europe map.

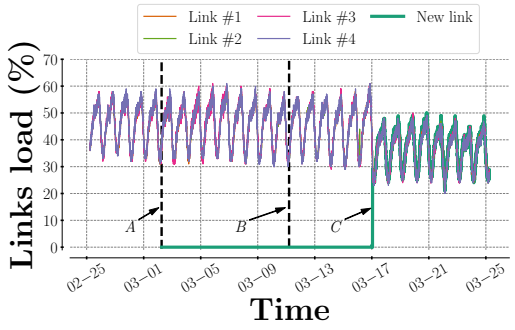


Figure 6: Links load towards AMS-IX over March 2022. A new link is added (*A*), reported in PeeringDB (*B*) and then activated (*C*).

of their imbalance values being lower or equal to 2 %. This low imbalance strengthens our hypothesis arguing that all parallel links have the same capacity.

Link upgrades. The last point that we explore in our dataset is when links are upgraded. A network operator typically decides to upgrade a link when its load increases and when technical constraints (e.g., routers ports) and cost constraints are met. Our dataset reveals such updates as abrupt changes in the link load. Figure 6 shows the effects of the addition of a link connecting AMS-IX on the loads of other links toward this peering. We observe that the new link was first added (arrow *A* in Figure 6), but not yet used. Then, PeeringDB [32] was updated (arrow *B* in Figure 6) nine days later, announcing a new link increasing the total capacity from 400 Gbps to 500 Gbps. Considering the four existing links before this upgrade, we can conclude that each link has a capacity of 100 Gbps. Finally, the link was activated two weeks after its addition and traffic was rapidly spread among all parallel links (arrow *C* in Figure 6), reducing the load of each link. The decrease in load observed in our data matches the added capacity reported in PeeringDB.

6 DISCUSSION

This paper introduces and presents an overview of the OVH Weather dataset. It did not attempt at analyzing in depth its characteristics and evolution over almost two years of collected data. Rather, we have illustrated how the available data could be used and hope for future research in that direction.

Several network operators, ranging from small ISPs to larger CDNs, also publish similar maps reporting the status of their network infrastructure. However, many of them provide a rasterised image of the map, for which the techniques developed in this work cannot be directly applied. One other French cloud provider, Scaleway, provides an SVG weather map of its backbone network [4]. While the network size is inferior compared to the one of our dataset, researchers could compare the collected data to understand the differences that could exist between the two networks.

A concern regarding any dataset is its validity. As discussed in Section 4, we have introduced a number of checks to ensure the correctness of the data extracted from the SVG images. Researchers could further validate the extracted data and even improve our algorithm. However, when considering the weather map as a view on a backbone network, the question of the extent of the network reported in the map is also important. When reaching to OVH after this work, they confirmed that the OVH Network Weathermap contains a very large part of their backbone network.

We argue that a key interest of our dataset is to combine it with other sources of information that can be undertaken in future works. For example, using PeeringDB [22] and BGP collectors, one could observe the evolution of peering links and their congestion [11]. Using `traceroute` measurements, researchers could correlate the utilization of MPLS [13, 48] or SRv6 tunnels [29] with the evolution of routing and link loads. Since OVH is a cloud provider, researchers could also install measurement servers to collect active measurements and correlate them with the link load in real-time, BGP events and other sources of information. Finally, OVH also reports planned maintenance events and the failures happening in their network in a dedicated website [2]. These events could give insights on the purpose of some modifications of their network and on their choices to mitigate failures. This source of information can then be used to augment our dataset.

ACKNOWLEDGMENTS

We thank the network infrastructure team of OVH for the interactions we had after this work. The work of Maxime Piraux was partially supported by the NGIPOINTER programme with funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement no 825354. Louis Navarre is an F.R.S.-FNRS Research Fellow.

REFERENCES

[1] 2022. Network Weathermap: Open Source Network Visualisation. (May 2022). Retrieved May 18, 2022 from https://www.network-weathermap.com

[2] 2022. OVHcloud: Network & Infrastructure Status. (May 2022). Retrieved May 18, 2022 from https://network.status-ovhcloud.com

[3] 2022. OVHcloud Network Weathermap. (May 2022). Retrieved May 18, 2022 from http://weathermap.ovh.net

[4] 2022. Scaleway Netmap. (Sep 2022). Retrieved Sep 12, 2022 from https://netmap.scaleway.com

[5] P. Ambati, D. Irwin, and P. Shenoy. 2020. No Reservations: A First Look at Amazon's Reserved Instance Marketplace. In *Proc. 12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*. USENIX.

[6] T. Benson, A. Akella, and D. A. Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/1879141.1879175

[7] RIPE Network Coordination Center. 2010. Atlas. (2010). See https://atlas.ripe.net.

[8] Y. Cheng, Z. Chai, and A. Anwar. 2018. Characterizing Co-located Datacenter Workloads: An Alibaba Case Study. In *Proc. 9th Asia-Pacific Workshop on Systems (APSys)*. https://doi.org/10.1145/3265723.3265742

[9] K. Cho, K. Mitsuya, and A. Kato. 2000. Traffic Data Repository at the WIDE Project. In *Proc. USENIX Annual Technical Conference*. USENIX. https://doi.org/10.5555/1267724.1267775

[10] kc claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. 2009. Internet Mapping: from Art to Science. In *Proc. IEEE Cybersecurity Application and Technologies Conference for Homeland Security (CATCH)*. https://doi.org/10.1109/CATCH.2009.38

[11] Amogh Dhamdhere, David D Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky KP Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C Snoeren, and Kc Claffy. 2018. Inferring persistent interdomain congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 1–15.

[12] B. Donnet and T. Friedman. 2007. Internet Topology Discovery: a Survey. *IEEE Communications Surveys & Tutorials* 9, 4 (December 2007), 2–15. https://doi.org/10.1109/COMST.2007.4444750

[13] Benoit Donnet, Matthew Luckie, Pascal Mérindol, and Jean-Jacques Pansiot. 2012. Revealing MPLS tunnels obscured from traceroute. *ACM SIGCOMM Computer Communication Review* 42, 2 (2012), 87–93.

[14] C. Dovrolis, K. Gummadi, A. Kuzmanovic, and S. D. Meinrath. 2010. Measurement lab: Overview and an invitation to the research community. *ACM SIGCOMM Computer Communication Review* 40, 3 (2010), 53–56. https://doi.org/10.1145/1823844.1823853

[15] Andrew D Ferguson, Jordan Place, and Rodrigo Fonseca. 2013. Growth analysis of a large isp. In *Proceedings of the 2013 conference on Internet measurement conference*. 347–352.

[16] S. Gay, P. Schaus, and S. Vissichio. 2017. *REPETITA: Repeatable Experiments for Performance Evaluation of Traffic-Engineering Algorithms*. cs.NI 1710.08665. arXiv.

[17] P. Gill, C. Diot, L. Y. Ohlsen, M. Mathis, and S. Soltesz. 2022. M-Lab: user initiated Internet data for the research community. (2022), 34–37 pages. https://doi.org/10.1145/1823844.1823853

[18] K. Keys. 2010. Internet-Scale IP Alias Resolution Techniques. *ACM SIGCOMM Computer Communication Review* 40, 1 (January 2010), 50–55. https://doi.org/10.1145/1672308.1672318

[19] K. Keys, Y. Hyun, M. Luckie, and kc claffy. 2013. Internet-Scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Transactions on Networking* 21, 2 (April 2013), 383–399. https://doi.org/10.1109/TNET.2012.2198887

[20] S. Knight, X. N. Hung, N. Falkner, R. Bowden, and M. Roughan. 2011. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications* 29, 9 (October 2011), 1765–1775. https://doi.org/10.1109/JSAC.2011.111002

[21] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on networking* 2, 1 (1994), 1–15. https://doi.org/10.1109/90.282603

[22] A. Lodhi, N. Larson, A. Dhamdhere, C. Dovrolis, and kc Claffy. 2014. Using peeringDB to understand the peering ecosystem. *ACM (SIGCOMM) Computer Communication Review* 44, 2 (2014), 20–27. https://doi.org/10.1145/2602204.2602208

[23] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. 2022. Inferring Link Weights Using End-to-End Measurements. In *Proc. ACM SIGCOMM Workshop on Internet measurment (IMW)*. ACM. https://doi.org/10.1145/637201.637237

[24] A. Marder. 2020. APPLE: Alias Pruning by Path Length Estimation. In *Proc. Passive and Active Measurement Conference (PAM)*. https://doi.org/10.1007/978-3-030-44081-7_15

[25] E. Marechal, P. Mérindol, and B. Donnet. 2022. ISP Probing Reduction with Anaximander. In *Proc. Passive and Active Measurement Conference (PAM)*. https://doi.org/10.1007/978-3-030-98785-5_20

[26] P. Mérindol, V. Van den Schriek, B. Donnet, O. Bonaventure, and J.-J. Pansiot. 2009. Quantifying ASes Multiconnectivity Using Multicast Information. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/1644893.1644937

[27] Giovane CM Moura, Ricardo de O Schmidt, John Heidemann, Wouter B de Vries, Moritz Muller, Lan Wei, and Cristian Hesselman. 2016. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In *Proceedings of the 2016 Internet Measurement Conference*. 255–270.

[28] S. Orlowski, R. Wessaly, M. Pióro, and A. Tomaszewski. 2010. SDNlib 1.0-Survivable Network Design Library. *Networks: an International Journal* 55, 3 (May 2010), 276–286. https://doi.org/10.1002/net.20371

[29] Victor-Alexandru Pădurean, Oliver Gasser, Randy Bush, and Anja Feldmann. 2022. SRv6: Is There Anybody Out There? *arXiv preprint arXiv:2205.04193* (2022).

[30] R. Pastor-Satorras and A. Vespignani. 2004. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press.

[31] V. Paxson and S. Floyd. 1995. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on networking* 3, 3 (1995), 226–244. https://doi.org/10.1109/90.392383

[32] PeeringDB. 2004–2022. The Interconnection Database. (2004–2022). see https://www.peeringdb.com.

[33] M. Peuster, S. Schneider, and H. Karl. 2019. The Softwarised Network Data Zoo. In *Proc. IEEE/IFIP 15th International Conference on Network and Service Management (CNSM)*. IFIP. https://doi.org/10.23919/CNSM46954.2019.9012740

[34] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In *Proc. ACM SIGCOMM*. ACM. https://doi.org/10.1145/2785956.2787472

[35] Y. Shavitt and E. Shir. 2005. DIMES: Let the Internet Measure Itself. *ACM SIGCOMM Computer Communication Review* 35, 5 (2005). https://doi.org/10.1145/1096536.1096546

[36] J. Sherry, E. Katz-Bassett, M. Pimenova, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. 2010. Resolving IP Aliases with Prespecified Timestamps. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/1879141.1879163

[37] H. Soroush, K. Sung, E. Learned-Miller, B. N. Levine, and M. Liberatore. 2013. Disabling GPS is Not Enough: Cellular location leaks over the Internet. In *Proc. Privacy Enhancing Technologies Symposium (PETS)*. https://doi.org/10.1007/978-3-642-39077-7_6

[38] N. Spring, R. Mahajan, and D. Wetherall. 2002. Measuring ISP Topologies with Rocketfuel. In *Proc. ACM SIGCOMM*. https://doi.org/10.1145/633025.633039

[39] N. Spring, L. Peterson, A. Bavier, and V. Pai. 2006. Using PlanetLab for network research: myths, realities, and best practices. *ACM SIGOPS Operating Systems Review* 40, 1 (2006), 17–24. https://doi.org/10.1145/1113361.1113368

[40] Staff, RIPE NCC. 2015. RIPE Atlas: A global Internet measurement network. *Internet Protocol Journal* 18, 3 (2015).

[41] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker. 2003. In Search of Path Diversity in ISP Networks. In *Proc. ACM Internet Measurement Conference (IMC)*. ACM. https://doi.org/10.1145/948205.948247

[42] Caida. 2002–2022. The UCSD Network Telescope. (2002–2022). https://www.caida.org/projects/network_telescope/.

[43] Caida. 2010–2022. Macroscopic Internet Topology Data Kit (ITDK). (2010–2022). https://www.caida.org/catalog/datasets/internet-topology-data-kit/.

[44] RIPE. [n. d.]. RIPE RIS, Routing Information Service. ([n. d.]). See https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[45] P. Tune and M. Roughan. 2017. Controlled synthesis of traffic matrices. *IEEE/ACM Transactions on Networking* 25, 3 (2017), 1582–1592. https://doi.org/10.1109/TNET.2016.2639066

[46] Steve Uhlig, Bruno Quoitin, Jean Lepropre, and Simon Balon. 2006. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Computer Communication Review* 36, 1 (2006), 83–86. https://doi.org/10.1145/1111322.1111341

[47] University of Oregon. [n. d.]. Route Views, University of Oregon Route Views Project. ([n. d.]). See http://www.routeviews.org/routeviews/.

[48] Yves Vanaubel, Pascal Mérindol, Jean-Jacques Pansiot, and Benoit Donnet. 2017. Through the wormhole: Tracking invisible MPLS tunnels. In *Proceedings of the 2017 Internet Measurement Conference*. 29–42.

[49] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. 2005. Network Anomography. In *Proc. ACM Internet Measurement Conference (IMC)*. ACM. https://doi.org/10.5555/1251086.1251116