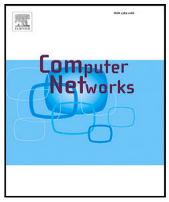
Computer Networks 221 (2023) 109523

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Workforce capacity planning for proactive troubleshooting in the Network Operations Center

Javier Ramos [a], José Luis García-Dorado [a,*], Javier Aracil [a,b]

[a] Department of Electronic and Communication Technologies, School of Engineering, Universidad Autónoma de Madrid, Spain
[b] Naudit High Performance Computing and Networking, Spain

ARTICLE INFO

Keywords:
Network monitoring
Proactive troubleshooting
Workforce capacity
Network Operations Center
KPI rankings
Time to analyze

ABSTRACT

Modern data centers require that a Network Operations Center is continuously monitoring network health, desirably in order to take proactive action before potential trouble occurs. In this paper, we contribute to the capacity planning of the workforce in charge. To this end, we have extensively analyzed, with real-world data, behavioral changes in a large server population in a data center. Our findings allow classifying such behavioral changes, which may be indicative of potential trouble, into relevance regions using a ranking mechanism. Then, the proposed methodology allows, together with an estimation of the time to analyze, assessing the workforce necessary to proactively tackle the behavioral changes observed. We conclude with a case study from a working data center, including a hands-on implementation of a traffic analysis solution to detect such behavioral changes and an estimation of the needed workforce to analyze them. Our results show that between 4 and 5 network managers are an adequate number for handling behavioral-changes analysis in a large enterprise data center.

1. Introduction

Nowadays, an increasing delocalization and digitalization of the workplace is taking place in all companies, which, unfortunately, is gaining momentum due to the COVID pandemic [1]. Thus, information technologies (ITs) have become a strategic asset for companies that must be fully operational at all times, to avoid stopping the company's activity. Such continuous operation relies completely on the networking infrastructure. If it fails, then several IT services, if not all, are affected and employees' work is interrupted. Moreover, with the wide adoption of remote work, not only the availability of services is important, but also their security [2]. In such a challenging scenario, the Network Operations Center (NOC) and the Security Operations Center (SOC) play a crucial role, as responsible for keeping the network healthy and secure.

Precisely, due to the COVID pandemic, the enterprise ITs departments have been stressed to their limits and so has the NOC. For example, as remote working was imposed in many countries, the NOC had to provide remote access for all company employees, all of a sudden. This is very challenging, as it involves many capacity planning and security issues [3].

In such a scenario, should the NOC fall short of workforce, then the ITs would be put at stake, possibly risking the company's revenue. This is the main driver and motivation of our research: to shed light on which workforce (in Full Time Equivalents—FTE—) is necessary in a NOC in order to preserve the network's health and security.

Needless to say, there are many different tasks carried out by NOC, and it would be unrealistic to pretend to model them all in terms of the necessary workforce. In particular, we focus on a specific area that we believe is carried out by any NOC and represents a significant share of the time: troubleshooting network operation. In this regard, we distinguish between reactive and proactive troubleshooting. And, within proactive troubleshooting, we distinguish between short-term and long-term analysis.

Reactive troubleshooting is a response to an immediate and pressing incident that has to be taken care of to resume a given service operation. To this end, the NOC/SOC vanguard is required to act swiftly and decisively because every minute counts. If the number of reactive troubleshooting incidents increases, then the data center capacity may be overflown, with serious business consequences as service continuity is compromised. Take as an example an e-commerce web portal that slows down, making customers desist from their purchases and consequently, severely affecting the business revenue.

Actually, to prevent NOC exhaustion, proactive troubleshooting is in order [4]. As such, the aim is to take action before a possible incident requiring reactive troubleshooting happens. Ultimately, the aim is to avoid reactive troubleshooting at all. However, this is very hard to

* Corresponding author.
E-mail addresses: javier.ramos@uam.es (J. Ramos), jl.garcia@uam.es (J.L. García-Dorado), javier.aracil@uam.es (J. Aracil).

https://doi.org/10.1016/j.comnet.2022.109523
Received 9 May 2022; Received in revised form 29 September 2022; Accepted 9 December 2022
Available online 13 December 2022

accomplish because one has to "guess" what is going to happen next. Even though we cannot reliably predict the future, we note that the past may give a hint for potential threats. For example, a sustained traffic increase in a given server, which never before showed such behavior, is worth being investigated, as it may finally result in a reactive troubleshooting incident. Generally speaking, proactive troubleshooting is based on detecting behavioral changes on the network and investigating the root cause before it is too late [5].

1.1. Proactive troubleshooting

Proactive troubleshooting deals with behavioral changes at different time scales [6]. Short timescales are relevant for anomaly and attack detection, where a burst of the measurements may be related to flash crowds, Denial of Service (DoS) attacks, or a port scan [7,8]. Short timescales are also relevant for detecting hosts and other network components that suddenly go down, for example, consider an unexpected reboot of a VoIP gateway. Particularly noteworthy is the role of short-term proactive monitoring in the field of cybersecurity, where the SOC team acts urgently as soon as disrupting activity happens.

On the contrary, changes at longer timescales (in the range of days or weeks) should be taken into account for network operation [8] such as network maintenance (e.g., to keep services working properly with no issues reported by final users), network fault and misconfiguration (e.g., software bugs and human errors, especially on several vendors' hardware) [9,10], and traffic engineering tasks for load balancing and capacity planning (e.g., inadequate resource configuration and traffic congestion) [11,12].

As such, long-term anomaly detection is specifically targeted toward identifying sustained host behavioral changes that may be worth investigating proactively. For example, a long-term anomaly happens if a given host that was issuing 1000 TCP connections per day progressively changes to 2000 connections per day. Such a behavioral change may be due to new software that has been installed in the host, or, for example, a printer or a physical access control with more use. These will not represent a problem as long as the number of connections does not saturate the infrastructure. Alternatively, connections may be issued by possible malware in the host. In the latter case, network administrators take proactive action, and not reactive as in the real-time counterpart.

In this light, proactive troubleshooting is key to avoiding risky situations. However, our working experience with many NOC/SOCs in recent years tells us that scarce resources are devoted to the analysis of changes that become stationary, whereas the shortest scales have received relatively much more attention [13–15]. Actually, it turns out that such a type of proactive detection and analysis activities is left unattended because of reactive and short-term interventions. Apart from obvious reasons such as expenditure cuts (which may very well be at the expense of the company's reputation and market position), we believe that there is a lack of workforce capacity planning methodology. As a result, nobody truly knows what workforce should be allotted to such an important task, and its corresponding effort is usually underestimated.

Precisely, this paper contributes to filling the gap of such workforce capacity planning, with a hands-on experience: a longitudinal analysis of a one-year worth of data in a real data center from a very large company. During that time, we proposed and implemented metrics to evaluate behavioral changes and supplied them to the NOC, which took care of finding the root cause. Thus, we are aware of how many changes happened and how long it took to deal with them. With that experience, we have been able to categorize behavioral changes based on their nature and model the time needed to inspect each change, giving rise to a methodology for workforce capacity-planning for proactive troubleshooting.

Thus, the main contributions of this paper are (i) a strategy to detect relevant behavioral changes based on a ranking mechanism in a given share of hosts, (ii) an approach to model the time and, consequently the workforce, needed to analyze each behavioral change, (iii) the determination of how many FTEs (Full Time Equivalents) are necessary to proactively deal with long-term behavioral changes.

The rest of the paper is organized as follows: in Section 2, we describe the analysis scenario, while Section 3 is devoted to the state of the art. Then, Section 4 explains our proposal and Section 5 presents a numerical example of our case study. Finally, Section 6 concludes the paper by providing some future lines of work.

2. Scenario description

The dataset used for model construction and analysis comes from the monitoring of several data centers' network segments of a multinational company for one year. Such a network provides connectivity between offices, connectivity to the management solutions, and other systems, as well as production centers, spread all over the world. To mention some of the services, in terms of office applications, there are deployed video conferencing and remote desktop services (RDP protocol) as well as shared real-time dashboards (VDI). Traffic from employees' workstations (mainly Microsoft Windows PCs) is also monitored. This traffic includes protocols such as NetBIOS, SMB, and LDAP/LDAPS used for enterprise login purposes and resource sharing. Additionally, a number of back-end systems, including databases and applications servers, some of them hosted on cloud infrastructures. Of relevance for management, HANA and SO99+ modules from SAP solutions [16] and other Supervisory control and data acquisition (SCADA [17]) applications for the management of machines and processes, which involves industrial protocols such as IEC 60870-5-104 or ModBus.

2.1. How traffic is monitored

We sniff data center traffic for all active VLANs in the core switches. The typical data center SDN star topologies with leaf switches connected through a central spine switch render themselves adequate for this purpose, as using a SPAN port in the spine switches is enough to capture the traffic that flows from clients to servers and also server-to-server traffic as long as such servers are located in different leafs.

We employ cost-effective solutions for traffic sniffing, based on commodity hardware. Specifically, our monitoring probe features two Intel Xeon Gold 6126 processors with 12 cores each and 192 GB of RAM. Regarding storage, the probe has two MegaRAID SAS-3 3108 RAID controllers with 36 SAS disks of 8 TB each. The disks are distributed in two RAID-0 volumes, providing thus a total storage capacity of 288 TB. The probe receives the traffic via a dual port Intel 82599ES 10 GbE card.

To capture and store traffic at 10 Gb/s rates and beyond, we employ our custom high-speed Linux driver [18], namely HPCAP. This driver provides low-latency reception paths from NIC queues to different user-space processing tasks, avoiding the overhead introduced by the standard Linux network stack. To this end, HPCAP implements circular reception buffers located in memory hugepages which are populated by a kernel-level polling thread that copies the packets from the NIC in a one-copy fashion. HPCAP presents a simple API that allows monitoring applications to fetch packets individually in a similar way to how libpcap does [19]. Additionally, the API provides functions to store traffic to disks efficiently by using page-aligned block writes. Storage functions also allow for transparent traffic capping, which significantly reduces the disk write throughput needed while increasing packet disk retention time until disks are full. This specific feature along with its programming model simplicity makes HPCAP a relevant approach for packet capture and storage compared to more complex capture engines such as DPDK [20].

Once traffic is stored in disk, a custom traffic analyzer (similar to the one described in [21]) loops through the captured traces, aggregating traffic in flows defined by the standard 5-tuple (source and destination

IP addresses and ports [22]) and reconstructing TCP sessions when possible. Besides the 5-tuple, each generated flow counts with more than 150 extra fields such as the number of TCP retransmissions, duplicate ACKs and 0-window, number of observed TCP flags, RTT, volumetry statistics, and even application-level information. Every 5 min, partial records for all the active flows are exported to disk. Simultaneously, another custom tool reads such partial flow records and aggregates them by IP address, selects only a subset of the 150 extra fields, and indexes the resulting records in Elasticsearch documents. Finally, Elasticsearch data is accessed by both Grafana dashboards and our workforce estimation tool, as detailed in Section 5.

2.2. Dataset

The dataset features almost 200 k unique IP addresses being more than 3000 of them servers and more than 700 k concurrent flows on average, peaking at 1 M concurrent flows. Typically, nearly 500 k enriched records are collected within a working day.

As noted before, proactive monitoring is based on modeling behavioral changes, namely, detecting situations that indicate a departure from the usual behavior related to networking. To this end, we follow an indirect estimation approach. By assessing the performance of communications between endpoints, we can tell if such a behavioral change has eventually happened.

Using the data provided by the monitoring system, we characterize each TCP or UDP connection through a Key Performance Indicators (KPIs) vector. These vectors are made up of a selection of the fields of the enriched Netflows that the monitoring system stores. That is, the measured KPIs accurately portray connection health and performance. Should any of the former parameters experience a significant change, it would be worth investigating why.

By default, such a set of KPIs comprises the following performance metrics per 5-minute interval:

- Received bytes.
- Sent bytes.
- Number of TCP duplicate ACKs.
- Number of concurrent connections.
- Number of TCP retransmissions.
- Number of TCP zero window announcements.

Nevertheless, such performance vector could be extended to incorporate more metrics depending on the particularities of the scenario under study. The anonymized dataset grouped by host for a year is freely available at [23].

3. Related work

Finding a correct definition of what a behavioral change constitutes is actually a challenge. The state of the art shows that changes can be modeled either in absolute or relative terms. The former happens whenever the metric value reaches or drops from a given threshold [24]. For example, the received/sent bytes metric falls to zero (or very small values) if a host suffers a network or power outage. However, things are more complicated when changes are not as simple as going to zero but arriving at some high or low values. And from this, the most significant weakness of threshold-based approaches arises, a formal definition of high or low tends to be arbitrary [8].

Typically, the variance of the time series itself is used to compute confidence bands [25,26] whereby the actual value of the measure under study should lie in under the normal network performance [27]. And to predict such a normal performance, there is an extensive body of research in the area of time series forecasting. For example, statistical approaches [28] have been applied in recent years and, nowadays, it is common to rely on machine or deep learning techniques [29]. However, often such statistical approaches assume parametric models to predict, for example, Gaussian distributions or others [30], and this modeling requires highly aggregated time series in order to show convergence as a natural consequence of the central limit theorem. Moreover, machine and deep learning techniques modeling tend to be intensive in processing power and samples.

While the above approaches have proven relevant for changes at short timescales [31–34], such as flash crowds or Denial of Service (DoS) attacks (where changes in load or the number of connections occur in a few seconds or minutes) or useful when a network component goes down (a router whose latency and number of retransmissions peak [35] immediately), we note that they are not able to detect progressive changes. That is, changes that do not occur suddenly, whose impact is not going to be in the range of some seconds or minutes, and that are maintained over time.

Certainly, the research community has paid attention to detecting attacks when they are already occurring but before having a significant impact on the operation. For example, the authors in [14] detect attacks using multiple machine learning models with the previously-labeled dataset of attacks by MAWILab. Similarly, in [15] LSTM neural networks are trained with manually-labeled web traffic anomalies, obtaining significant accuracy. In particular, they use Yahoo's Webscope S5 dataset. As another example, the authors in [13] present an intrusion detection system for network security based on deep learning models. The results are evaluated on self-supervised labeled traces, performing with an accuracy of nearly 100%. Finally, in [26], it is presented a real-time alerting system where the threshold of normality is estimated by exploiting also LSTM neural networks and validated with a dataset gathered from Facebook Prophet.

On the one hand, it becomes apparent that our aim of analyzing behavioral changes over time is different from such a set of systems targeted at detecting attacks and security breaches in real time. Moreover, we also remark that the performance evaluation is far different: First, leveraging labeled training datasets comes with significant limitations: the process of labeling requires an effort by network managers, the evaluation is limited to a set of well-known issues and the performance of dynamic or new threats is unknown. Second, by the time that evaluation is carried out, the identification of an attack behaves as a binary classification where the system detects the attacks that are known to be in the trace or not. However, the same does not apply when the aim is detecting behavioral changes. That is, the network's measurements, for sure, are changing. For example, consider the scenario of a loaded Gb/s link that increases the bandwidth by only 1 b/s. Certainly, there is a change there but not in the same sense that an attack is detected, here the point is being able to conclude if the change calls for its analysis.

Probably the difficulties described above have meant that the study of stationary changes has received much less attention than other types of changes, albeit its relevance in capacity planning among other tasks. Certainly, the authors in [6] focused on stationary changes by considering the bands for normal behavior as a vector of multiple thresholds instead of a scalar. Each threshold accounted for a 90-minute interval throughout the day. They assumed that the behavior of a large traffic multiplex in each of these intervals should behave as a multivariate Gaussian process. Then, a change from the regular performance was only considered relevant if it happened several times and at different time intervals. On the one hand, note the threshold-based problem remains, that is, instead of an absolute threshold as other works, there are several ones to be considered in this case, while a new parameter appears, i.e., how many intervals are enough to consider a change as a steady change. Moreover, note that applying this statistical approach requires a metric with a significant aggregation in order to converge to a model. This could be feasible as the authors were measuring bandwidth in a large traffic multiplex but not for individual hosts and other types of metrics as we are focusing on.

As a conclusion, proactive monitoring needs mechanisms to distinguish when a change is behavioral from a simple and natural variation in the operation of a network. And, once a change is considered behavioral, it needs mechanisms to give relevance to such changes. This way, the most relevant changes are analyzed first, and the less relevant ones can be analyzed later or, given that time for analysis is limited, not even be considered.

4. Proposal and methodology

In the scenario described in the previous section, hereafter, we explain how the use of KPI rankings, or a set of ordered IP addresses by a given KPI, is a relevant mechanism to detect changes without a modeling or training phase. Also, we apply the concepts of top and coverage to reduce the burden while the most relevant behaviors are still analyzed. As a metric to measure relevance, we use the difference in position of a given IP address in a ranking compared to another. And finally, those similar changes in terms of relevance are estimated to need similar times to be analyzed. After some analysis of previous events, this results in an estimation of the workforce. This factor is not typically considered, albeit there is a direct relation between the capacity of analysis and the number of changes that can be analyzed. In this paper, we pay attention to this issue.

4.1. Rankings

We believe that it is a better approach to model changes in relative terms rather than absolute terms for the case of detecting the long-term behavioral changes we are considering. Thus, the approach is to consider the relative magnitude of a metric from a given host with respect to the others, that is, to add the sense of order and relativity. Moreover, as in any real-world system where cost-effectiveness is key, only a limited number of hosts can be tackled, which implies that a prioritization scheme is in order. To combine these two concepts, we propose to use KPI rankings: KPI rankings are defined as sets of pairs of IP address — KPI value, ordered in descending order by such KPI value. More formally, let the number of hosts in the network be S and K the number of metrics under surveillance. We define the *vector of metrics (or KPIs)* as a vector $(X_1^i(n), \ldots, X_K^i(n)), 1 \leq i \leq S, n \geq 1$ such that each X_i component represents a metric (say, for instance, number of connections) in the stability interval n. Each metric X_j^i is ordered for each interval n, namely by the i super index, such that

$$X_j^{(1)}(n) \leq \ldots \leq X_j^{(S)}(n), \quad 1 \leq j \leq K \tag{1}$$

namely, a map $(k) \rightarrow i, k = 1 \ldots S, j = 1 \ldots K$ is established that shows that host i occupies position k in the ranking, for a given interval n and metric.

Then, in order to analyze relative changes, for a given time period based on the KPI values, we build rankings for a given time period based on the KPI values and compare them with previously calculated rankings. We consider that a *change* has occurred in a given metric for a given host if and only if its relative position in a ranking has actually changed in a given time. Let us refer to this time as the *stability interval* which will be analyzed in the next section.

In conclusion, we have a mechanism to infer changes based only on positions and where the relevance of a change can be easily measured as the difference in position between two ordered lists. Additionally, note that this approach comes with significant advantages: the ranking system is KPI agnostic (as long as a KPI is numeric-based and measurable), insensitive to units of measure, sensitive to abrupt and progressive changes, there is no need for modeling and learning phases, and new KPIs can be added as needed without changing the system. Additionally, as the comparison is relative, the ranking method is immune to the influence of variations in the value of the KPIs due to external data acquisition problems such as packet duplication, which is very common when a SPAN port is used [36].

Finally, the ranking approach is simple in terms of both calculation and operation and eliminates the need for performing training or more complex modeling such as the ones involved in statistical and deep learning methods, which allows for a more general analysis method and reduces the amount of data needed to operate.

4.2. Stability interval

To define a stability interval, we rely on qualitative analysis and measurements. As for qualitative reasoning, we note that in most enterprise data centers there is some degree of periodicity in the operation of a service [37]. The stability interval serves to filter out instantaneous variations in the metric under study that drive the host position in the ranking up and down continuously. Note that the stability interval must be large enough to capture the periodicity of events that occur during normal network operation that are strongly correlated to human activity. For example, some services are provided only during the daytime or during working days and not over the weekend. This way, one-week intervals should be stable enough to capture the labor/non-labor operation of the network. And one-month ranges are stable enough to capture other events, such as payroll processes.

Other stability intervals, although periodic, are not useful for proactive monitoring. For example, comparing one-year rankings, by definition, requires aggregating data for years and the conclusions drawn should intuitively be too old to be useful in a proactive approach.

Similarly, small periods such as one day will have other limitations, such as too much burstiness. Firstly, one day can be not enough to show period behaviors, for example, weekends tend to be different from working days [37,38]. And even, Fridays can be different given that human activity periods differ. Secondly, a one-day interval is more related to reactive monitoring approaches, as it is more useful in tasks such as alarm triggering during peak hours or short time periods that can detect spurious changes or deviations from normal behavior. In this scenario, the traffic characterization has a reduced time scope and is not significant enough to state that a behavioral change has occurred on the network and that such a change has altered the previously observed operational behavior.

4.3. Limiting burden: Top selection and coverage

Data centers are growing steadily, with a CAGR (Compound annual growth rate) of more than 3% [39]. As a result, a large data center features thousands of servers, each of which may, in turn, run virtual machines or containers.

In such a challenging scenario, one may attempt to account for all possible behavioral changes of all possible hosts in a data center. Clearly, such an approach leads to a daunting number of changes. Thus, one has to select the hosts that will be monitored for changes, such that the number of hosts gets reduced to a reasonable figure. To this end, a qualitative assessment of the server business relevance can be performed, in which only those hosts that are very relevant for a particular metric would be subject to behavioral change analysis. Such an approach has the advantage that a fully automated choice of the hosts can be performed, with no human intervention.

In this light, following Eq. (1), we define the *top* of a ranking as the first N hosts that

$$X_j^{(1)}(n) \leq \ldots \leq X_j^{(N)}(n), \quad 1 \leq j \leq K, 1 \leq N \leq S \tag{2}$$

Then, let us define the coverage of a metric according to a given top as the sum of the values of the hosts in the top makes with respect to the sum of the contribution from all the hosts in the sample. More formally, the coverage (C_j) of a metric (j) for a given top N is:

$$C_j = \frac{\sum_{t=1}^{N} X_j^{(t)}(n)}{\sum_{t=1}^{S} X_j^{(t)}(n)} \quad 1 \leq j \leq K, 1 \leq N \leq S \tag{3}$$

As an example, if top is equal to 1 and the metric is the number of received bytes, the coverage is directly the volume of bytes received by such top-1 host divided by the sum of the received bytes by the rest of hosts in the sample. Then, the coverage, assuming a top equal to 2 is the sum of the bytes received by the two most active hosts in terms of

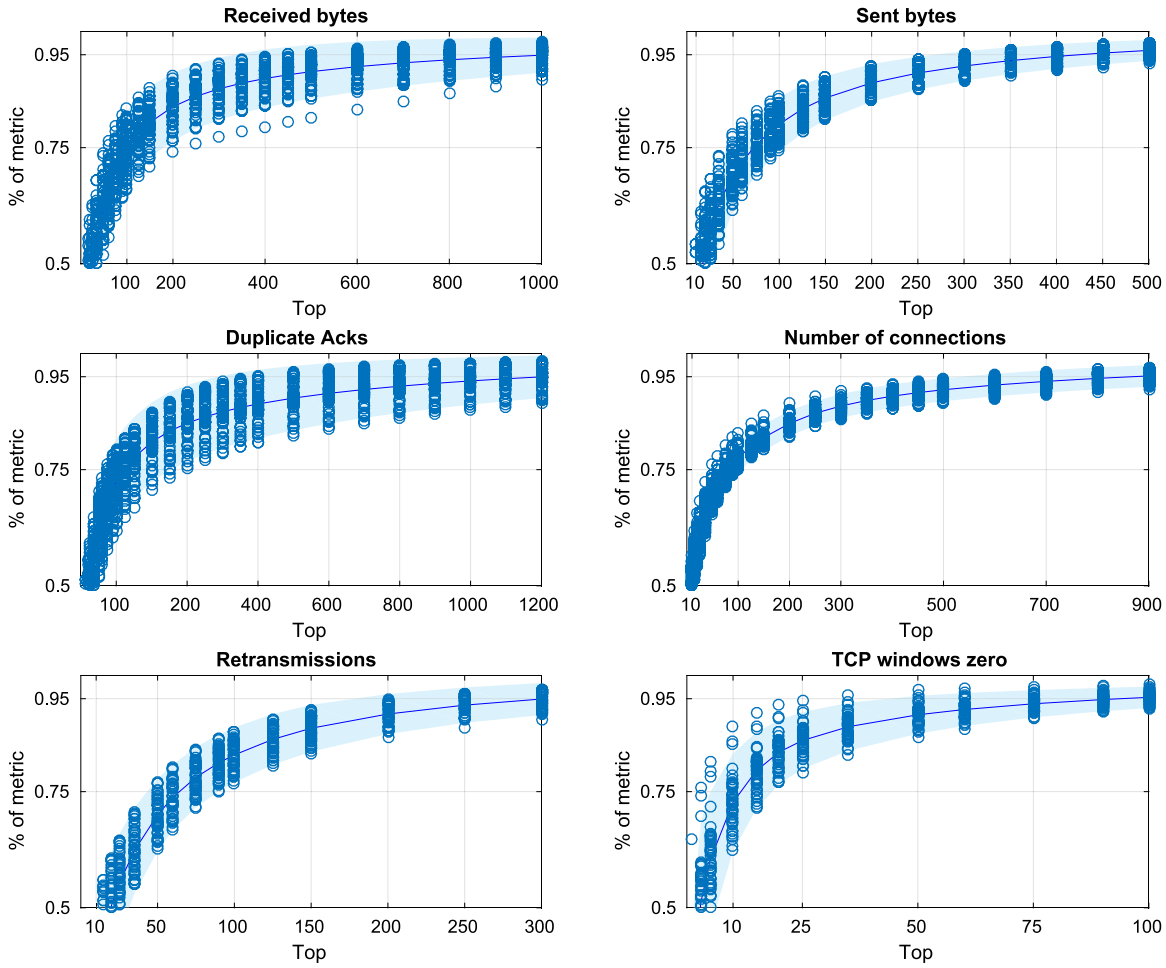


Fig. 1. Top hosts that provide a coverage of 95% for each metric.

received bytes divided by the total number of bytes received by all the sample. And so forth.

Fig. 1 illustrates the coverage for the set of different behavioral metrics as the top increases for a year. Each point in the graph represents data from one week's worth of traffic. This way, each top shows 52 samples that make up a whole year's worth of traffic. Also, the mean and 95% confidence interval are depicted.

We note the graph shows different tops until the metric spans 95% of the total. As it turns out, the host subset size varies between 100 and 1200 hosts, depending on the metric being considered, which, in the worst case, represents less than 0.5% of the total number of terminals. Moreover, not surprisingly, a small number of hosts accounts for the majority of the traffic, in bytes and flows. For the rest of the metrics, the same happens: a few hosts account for the most values of the metric. Such a Pareto-like effect allows us to discriminate the relevance of the hosts.

As a conclusion, this allows further optimization of the workforce, which should be focused on relevant behavioral changes, IP addresses in a different top for each metric, and not all of them. For the sake of completeness, a similar Pareto-like effect was found for coverage for a one-month interval. In particular, given that more data is aggregated, such a Pareto-like effect was even stronger.

4.4. Relevance of changes

We posit that the relevance of a given change can be measured by the number of positions that the corresponding metric has changed in a ranking compared with previous rankings. For example, if the top 1 host in number of connections drops to the last position in the ranking, this is for sure a very relevant change. Conversely, if it hops to the second position, perhaps, it is not worthy of further analysis. We further note that the relevance of a given change serves for workforce capacity planning as well, since it allows *rating* changes according to relevance, such that the most relevant ones are taken care of first up until the available workforce capacity.

More formally, a change in metric happens for a host i if $(k') \rightarrow k$ in interval $n+1$, and the change relevance can be measured by the difference of positions in the ranking, namely a *hop* (Δ), where $\Delta = |k' - k|$. As it turns out, the larger Δ is the more relevant the change is. In $\Delta = 0$ then no changes happen for that particular metric and no effort is required from the NOC team. Conversely, if Δ is large for a given host then a behavioral change has occurred for that particular host (the larger the worse) and action is required from the NOC team.

To illustrate the concept of relevance, Fig. 2 shows the distribution of the hops that occurred per metric for the year under monitoring, considering weekly rankings. In particular, such distributions are depicted as violin plots where a smoothed probability density function of the variable under study is drawn in the vertical axis overlapped with the average and interquartile range [40,41]. For the sake of easier visualization and comparison, the hops are shown on a logarithmic scale.

Several observations arise. Most of the metrics share a similar shape: A mode accounting for those hops near the mean and median. Another mode becomes apparent at the tail of the figures. And, in between these modes, a fraction of samples more or less numerous depending on the metric. Interestingly, we can relate these regions to different types of network issues.

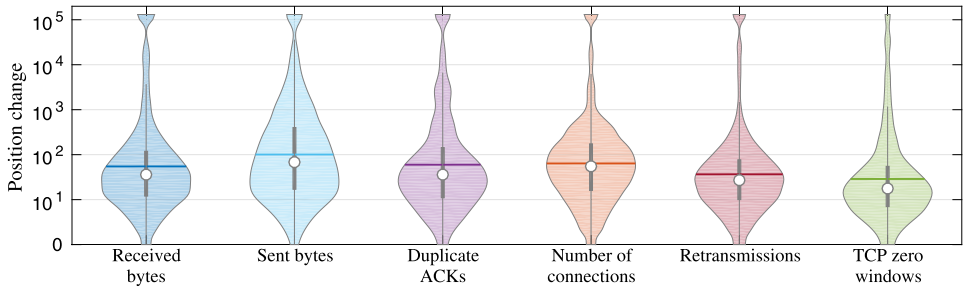


Fig. 2. Change relevance distribution for the set of metrics in logarithmic scale.

That is, the latter mode accounts for very long hops. These position changes portray the case of hosts whose relevance was so high to be included in the top but that, in the following period of time, they fall at the tail of the relevance ranking. Or vice versa, hosts that entered the top when they were in the tail in the previous period of time. Such hops, in the tens of thousands range, can be caused by either a dramatic behavioral change (such as a total resource exhaustion, for example, CPU or bandwidth) or due to manual intervention in the host, such as a switched-off machine, maintenance tasks, planned systems disconnections or load balancing across several VM instances. These are issues to be addressed with urgency, as they are measurements that went from the top to zero. However, as noted, the reason behind such a change can be trivial or, likely, the reactive monitoring was able to detect them. That is, proactive monitoring searches for problems that are starting to occur and may be critical in the distant future, for instance, a link saturated due to a sustained increment of the number of users of a service. If a problem took place and its impact was abrupt and large, the reactive monitoring would probably have already reported it.

The samples on the first mode account for small hops, changes in the range of the top (between 100–1000 positions), and probably they represent spurious changes, which do not point to behavioral changes of relevance. In other words, the simple and inherent variance of the metrics' time series may cause certain changes in the rankings. In this scenario, sometimes, even an expert analyst would not be capable of finding an explanation for such subtle changes, although the analysis time was unlimited.

Between these two modes, we have a relevant region. The hosts of such a region traversed from a top position to a position at the end of the distribution. Not to the tail, but a position far from the latest values. These hosts deserve attention as they pointed out changes that are not sudden, but changes that are starting to happen, and such changes are difficult to be detected by a reactive-monitoring system. Unfortunately, these issues may demand significant time. Note that the correspondent analysis here is focused on understanding the root cause of why something is working differently, not a host that stopped or a recently launched VM that started to transmit. Actually, the analysis of these types of changes can consist in the search for an explanation for a problem that perhaps does not exist. For example, consider the case of a link going to a quarter of utilization simply because the applications behind them are being less used. Or, similarly, consider a period of holidays that impacts the use of certain services on the network. That is, the time required to rule out that the change represents a problem is often harder than finding a problem when it exists.

Let us formally relate the types of changes, regions in the distribution of relevance, and analysis times:

- *Region IV — Urgent changes, moderate analysis times*: That is, these samples are hosts that were in the top for a week and went to the tail in the following week (or vice versa, they arrived at the top from the tail). The analysis cost is not high as some changes can be easily analyzed.
  Let us include a host in this region if the position change in ranking is more than 90% of the total host population. As an illustrative example of an event that fits in this region, consider the case of a host that has been switched off after a long working period.
- *Region III — Relevant changes, high analysis times*: This group comprises significant hops in the ranking in terms of size, but such hops did not cause the host to fall into the tail of the distribution. They would require detailed analysis.
  We include a host in this region if a change was longer than the top, but smaller than 90% of the total host population. To exemplify events in this region, consider the case of a link whose utilization is increasing due to new users being added to a service.
- *Region II — Low relevant changes, very high analysis times*: This group brings us changes of position inside the top. Such changes are far less relevant than the previous ones, as the host is still within the top length. In the case of being analyzed, such changes demand a detailed study.
  We define such a region when the position change in ranking is more than 10% but less than 100% of the top. One example of an event in this region is the reduction of traffic due to regional holidays that produce a subtle reduction in the number of users of a service during one or maybe two days.
- *Region I — Irrelevant changes, no analysis*: This group includes hosts that changed very little or even did not change at all in the ranking. These changes are not studied.
  We define such a region when the position change in ranking is less than 10% of the top. The type of event that is located in this region is produced typically by the variation of use due to the very nature of the services. As an example, the duration (and thus the amount of traffic generated) of the daily remote desktop connections is not constant throughout the week and depends on employee utilization and workday distribution.

Let us illustrate these regions of changes with an example. Fig. 3 shows the above-mentioned regions, for the received-bytes metric assuming a top of 1000 hosts, for a representative week of the year. In particular, the figure divides the probability density function of hops into the previously described regions.

In conclusion, we note that the above taxonomy allows for classifying changes into relevant regions or categories, such as the most relevant changes must be attended to first. And, importantly, we have explained that each region represents a different set of issues, and, coherently, the time for analysis will be different between regions and similar between samples inside the same region. This serves to quantify the NOC workforce in the following section.

4.5. Quantifying NOC workforce: Regions and TTA

Once behavioral changes have been categorized, the required workforce would be the addition of the number of changes in each region multiplied by the time to analyze each type of issue.

Let n_I, n_{II}, n_{III} and n_{IV} be the number of hosts in the defined categories, then the workforce (W) is measured in time for a given stability interval:

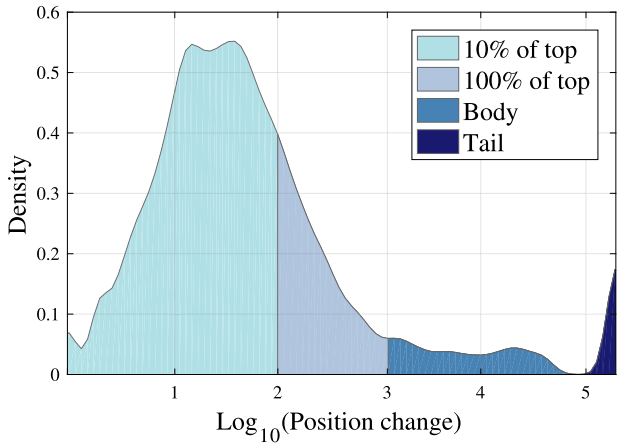


Fig. 3. Example of an illustrative week for received-bytes metric where the change relevance regions are highlighted.

$$W = \alpha_I \cdot n_I \cdot TTA_I + \alpha_{II} \cdot n_{II} \cdot TTA_{II} + \alpha_{III} \cdot n_{III} \cdot TTA_{III} + \alpha_{IV} \cdot n_{IV} \cdot TTA_{IV} \quad (4)$$

where the α_i values range between 0 of 1 depending on the desired attention to each category i, and TTA_i represents the time to analyze a change in region i.

A conservative approximation would be setting $\alpha_{III} = \alpha_{IV} = 1$ and $\alpha_I = \alpha_{II} = 0$ whereby only the relevant changes are studied. The increase of α_{II} will provide a more detailed study of a fraction of the changes considered only moderately relevant. Also, the stability intervals may focus only on weekly changes or also include monthly data.

The TTA, time to analyze, is simply a variation of the well-known term TTR, time to repair. While TTR focuses on the time required to repair a machine or system, we pay attention to the time required to provide an analysis of a behavioral change. TTR has received much attention for its relevance in the measurement of systems' performance, design, and reliability [42] for more than three decades. In this way, we can borrow all the efforts dedicated to characterizing such a metric. In particular, the Mean Time to Repair (MTTR) metric represents the average time to repair a system considering the last N reparations. Similarly, MTTA is the average time a NOC needs to analyze a behavioral change.

4.5.1. Limitations

However, a simple mean often may not be a good approximation for heavy-tailed shapes that the TTR and TTA processes can follow. In this light, they can be posed as a distribution function, not a scalar, following, typically, a log-normal or exponential distribution [43]. The implication of Eq. (4) is that each term $n_i * TTA_i$ is a convolution of n_i terms. In the case of assuming an exponential distribution, the sum will follow a Gamma distribution and, in the case of considering a log-normal distribution, the sum can be estimated with approximation methods [44].

Anyhow, regardless of considering TTA as a scalar or stochastic process, the key is to have a significant number of samples to fully capture the variability of the process. That is, it is unlikely that all network managers are equally efficient (or even that a given analyst works the same way every day), but it makes sense that after the aggregation of enough samples the distribution converges. In other words, the best, worst and regular analysts/working days are offsetting the estimation. Actually, the authors in [43] suggest measuring in the range of hundreds of samples to ensure the significance of MTTR estimate but, at the same time, they state that in practical cases this number may be in dozens rather than hundreds. In the case of our work, we have used a number of samples an order of magnitude above hundreds on our hands-on working experience with operations centers, this provides us with a significant estimate with narrow confidence intervals.

5. Case study

In this section, we describe how to numerically estimate the workforce for the NOC of the multinational company described in Section 2 by means of the methodology explained in the previous section.

5.1. Operation in the NOC

We have developed a visual proactive-monitoring system through standard and custom-made modules of Grafana [45]. Fig. 4 illustrates such a monitoring system whose details are the following:

At the back-end side, each Monday early morning, aggregation metrics are calculated and, automatically, the top and region thresholds are fixed. With this, the set of IP addresses per region and metric are estimated. According to the parameter α, the system elaborates a list of IP addresses to analyze ordered by priority (left part of Fig. 4).

Then the regular operation of a network manager is to click the first one that has not been previously analyzed by another coworker. Then, a new dashboard came up to the manager (right part of Fig. 4). Inside, the managers can find the time series of the six-default metrics for the last week, two weeks ago, and for both the current month and year. Additionally, time series for on-demand time intervals can also be depicted. The metric or metrics, we note that more than a metric can be of relevance for analysis, that made the address of relevance are marked in red.

Interestingly, thanks to the capacity of Grafana filters and variables, such time series can be tuned to depict time series from/to a specific set of IP addresses, port numbers, or services (e.g., SAP, DNS or teleconference applications). Finally, the managers have available buttons to download Netflows-like records and packet traces to further inspect particular time intervals, for example, those that the time series suggest are abnormal. In these tasks, well-known network analyzers such as Wireshark can help. Packet traces may demand a significant amount of storage capacity, for default we store traces for one week, while flow records are stored for one month.

As standard Grafana plugins do, we have developed a back-end service that, after receiving the parameters chosen by the manager, retrieves the PCAP file or files and concatenates them. Finally, the system serves the filtered trace using HTTP for its download. As an alternative, the system can open the trace and forward the output of Wireshark to an HTML5 interface. This allows the trace to be viewed, avoiding its download and without the need of having Wireshark installed.

The dashboard also shows specific KPIs for certain protocols. In the case of TCP connections, the system tracks the number of SYN and RST flags detected, as well as latency measurements. For HTTP, DNS, or SAP packets, the system also tracks the number of erroneous notifications. Together with all this data, network managers have access to the inventory, operational, and historical databases. In particular, the latter contains, for each of the previous issues, a report with the analysis carried out. This database can be filtered by the metrics that provoked the behavioral change, metric values, relevance, and any keywords. This way, network managers have an easier time looking for problems, finding the root cause of the behavior change, or simply determining that a change does not have relevance (e.g., a switched-off machine or a planned upgrade).

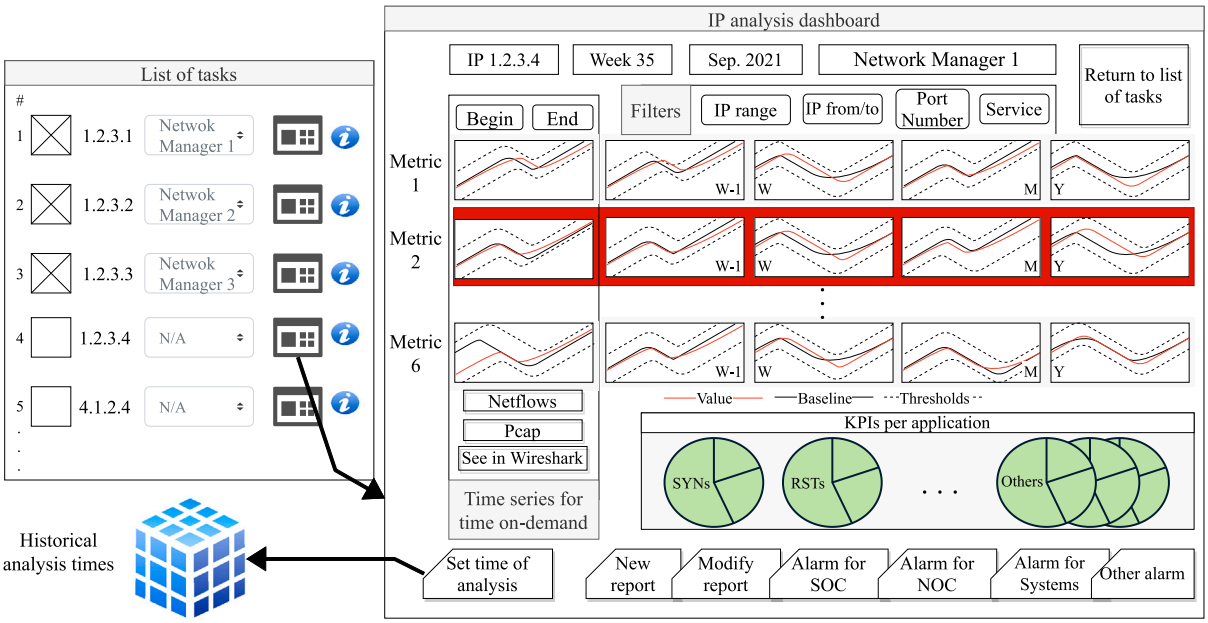


Fig. 4. Monitoring system and dashboards.

To provide feedback, network managers must create a report explaining their findings as introduced before and, if necessary, trigger an alarm. Both outputs must be treated subsequently by one of the support departments of the infrastructure. They are receiving a formal report describing the issue. For example, a machine saturated or a virtual machine not working properly must be handled by the systems department and an increase in the number of retransmissions must be forwarded to the network department. Finally, such a department will take opportune measures to solve the issue. Note that time required to solve an issue can be far different from the time to analyze it. For example, consider the case of a saturated link. The time to upgrade it can be only a few minutes for a virtualized connection, but it can be days in the case of a physical upgrade, whereas the time to find the saturated link does not depend on the type of link.

As the final step, the network managers are asked to mark the time or time intervals that they used for each analysis. This activity track and ticketing systems are very common in companies to assign dedication times to a particular project; hence, it should not entail any additional burden. These times are received by the monitoring system that upgrades TTA estimations per region and several distribution charts are formed. This provides the information needed for the task of workforce capacity planning.

5.2. Example of operation in the NOC

Let us detail a successful use case of our system for a recurring issue with a monitoring and management software solution [46]. Such software is in charge of monitoring systems, applications, and network devices to gather information such as CPU and memory usage, temperature, event logs, etc. The software works in a client–server fashion, with a central server that receives data from thousands of agents installed at different hosts or network devices. Periodically, the agents establish a TCP connection to the server, transfer the information collected, and close the connection. Although the application closes the connection when data is acquired, the TCP connection stays active at the operating-system level for several seconds to ensure complete data transmission between server and client. The closed connection stays active, typically, for 1 min, and therefore the resources remain allocated and cannot be used by new connections. This should not be a problem as the monitoring software schedules the connection establishment from agents evenly spaced in a way that does not exhaust the available resources, and only hundreds of connections are active at the same time.

Focusing on the chronology of the problem, first, our system detected increments in the number of connections for the server. This resulted in an increase in the ranking position of the IP address associated with such a metric. During the first month analyzed when the problem was present, the increase was subtle, and the change in the ranking position was assigned to the Region II of the relevance distribution. At that moment, the event was reported to the NOC workers that started studying the problem with low priority as other events were more relevant at that time. In the next month, our system detected a massive change in the position of the IP address of the server in the number of connections of that server. Such position change was assigned then to the Region III of the relevance distribution and was reported as urgent to the NOC workers along with an estimation of 45 min needed to diagnose the problem. In parallel, during the last days of that second month, a few tickets were opened by users complaining about random failures in accessing the service. While there was no full-service outage and the service continued working, some instability symptoms were present.

After receiving the tickets and our recommendation of inspecting the server, the NOC workers analyzed the problems and found the root cause. As it turned out, during a software update, the developers of the tool reduced the agent data-gathering intervals without warning about it. That broke the even distribution of requests previously observed. This resulted in thousands of concurrent connection attempts that, in some cases, went unanswered due to exhaustion of server resources as closed connections remained active for 1 min as stated before. This aggravated, progressively, the problem, as agents that were not able to establish a connection sent connection retries with intervals of less than a minute, thus generating more load for the server.

Once the root causes were analyzed, the NOC team decided to increase the maximum number of connections the server can handle and reduce the time period that connections are still active after being closed (from 60 s to 20 s). These changes solved the problem, and all agents were able to transfer data without any issues. All the troubleshooting and solving the problem took effectively 1.5 h from the NOC team, which deviates 45 min from our initial estimations. This exemplifies a case where the analysis needs extra time from the network managers in contrast to cases that are trivially solved.

5.3. Workforce figures in the NOC

Once described how we propose network managers receive their tasks and illustrate with an example of the operation, let us provide some estimations of the workforce in the NOC.

By default, we set α parameter in such a way that the changes from regions I and II receive full attention while the other regions are not analyzed, i.e., $\alpha_{III} = \alpha_{IV} = 1$ and $\alpha_I = \alpha_{II} = 0$. Similarly, weekly analysis is provided by default while monthly analysis is only optional. As a good trade-off between simplicity and applicability, we use the mean of the TTA process to estimate analyzing times.

As coverage, we parametrize the top to span 95% of each metric. That is, tops are set to 1000, 500, 1200, 900, 300, and 100 hosts for the six metrics studied by default as Fig. 1 showed. In particular, we are considering the last two hundred samples to estimate the mean. By the end of 2020, the MTTA was measured at 45 and 12 min for regions III and IV, respectively. For its part, category II averages more than one hour. This result does not come as a surprise. As previously noted, it is easier to find a trivial problem (category IV) than to find the root cause of a relevant problem (category III). And what is more, it is even more time-demanding to rule out that a problem exists given the inherent variance of communications behavior (i.e., category II).

Table 1 shows the average number of IP addresses in each of the regions per metric for the 52 weeks of the year 2020. The last row depicts the number of unique ($\cap$) addresses for each of the regions considering all the metrics, that is, the intersection of each of the columns of the table. It becomes apparent that it is common for a change to involve more than one metric at the same time. For example, consider a payment gateway that increases the number of transactions as it covers more payment points, causing metrics such as Received bytes, Sent bytes, and Number of concurrent connections to increase at the same time.

This way, following Eq. (4), the average per-week workforce (W) in minutes is $216 \cdot 45 + 98 \cdot 12$ resulting in 10,896 min, or equivalently, 181 h, less than 23 working days (assuming 8 h per working day), and consequently, between 4 and 5 FTE network managers (assuming 5 working days per week). In the case of 5 FTE managers to get balanced capacity and dedication, α_{II} may be set to 0.03 to obtain full occupancy while including a few of the most relevant changes from category II. In case of having a smaller budget, α_{III} may be reduced to 0.86 to set the number of FTE network managers to 4 (i.e., 86% of the relevant changes). Similarly, if the maximum number of available network managers is 3 FTE, α_{III} should be fixed to 0.62, for 2 FTE network managers α_{III} takes on 0.37, and, finally, for lower funds, α_{III} is 0.12 for a single FTE network manager. On the other hand, if the budget is sufficient to support even low-relevant changes (i.e., category II, with an estimated MTA of 69 min), the analysis demands $536 \cdot 69 + 216 \cdot 45 + 98 \cdot 12$ a week, which translates in, roughly, 20 FTE network managers.

To put figures into perspective, for the analyzed year in the multinational company NOC under study, we found that roughly 60% of the reported issues needed attention from the support departments of the infrastructure and within those, 15% required corrective measures by data-center technicians.

6. Conclusions

Through extensive analysis of real-world data center data, we have come up with a methodology that allows identifying behavioral changes and quantifying the workforce required in a NOC to proactively analyze them. Such changes are classified into change relevance regions based on KPI rankings which, combined with a TTA estimation, provides a final estimation of the NOC workforce.

Future work includes the identification of relevant metrics beyond traffic, such as those coming from logs or telemetry APIs, to identify hosts that not only show changes at the network level but also at the systems/application level, which could be classified as of high priority for the NOC/SOC workforce.

Table 1
Number of IP addresses falling into regions in weekly average per metric.

		Regions			
		I	II	III	IV
Metrics	Received bytes	814.65	183.71	56.14	43.68
	Sent bytes	283.70	204.29	97.01	19.91
	Duplicate ACKs	1005.46	212.60	76.80	53.87
	Number of connections	615.29	281.83	43.68	25.47
	Retransmissions	166.35	132.06	20.17	9.34
	TCP zero windows	41.65	55.07	18.79	2.58
	$\cap$	1580.63	536.28	215.69	97.55

CRediT authorship contribution statement

Javier Ramos: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization. **José Luis García-Dorado:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization. **Javier Aracil:** Conceptualization, Methodology, Validation, Formal analysis, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research has been partially funded the Ministry of Science and Innovation of Spain through Project AGILEMON under Grant AEI PID2019-104451RB-C21 and by Naudit High Performance Computing and Networking under art. 83. projects.

References

[1] F. Almeida, J. Duarte Santos, J. Augusto Monteiro, The challenges and opportunities in the digitalization of companies in a post-COVID-19 world, IEEE Eng. Manage Rev. 48 (3) (2020) 97–103.
[2] J. Grimm, Securing the remote workforce in the new normal, Comput. Fraud Secur. 2021 (2) (2021) 8–11.
[3] M. Candela, V. Luconi, A. Vecchio, Impact of the COVID-19 pandemic on the Internet latency: A large-scale study, Comput. Netw. 182 (2020) 107495.
[4] G. Nguyen, S. Dlugolinsky, V. Tran, A. López García, Deep learning for proactive network monitoring and security protection, IEEE Access 8 (2020) 19696–19716.
[5] G. Aceto, A. Botta, P. Marchetta, V. Persico, A. Pescapé, A comprehensive survey on internet outages, J. Netw. Comput. Appl. 113 (2018) 36–63.
[6] F. Mata, J.L. García-Dorado, J. Aracil, Detection of traffic changes in large-scale backbone networks: The case of the Spanish academic network, Comput. Netw. 56 (2) (2012) 686–702.
[7] L.F. Carvalho, S. Barbon Jr., L. de Souza Mendes, M.L. Proença Jr., Unsupervised learning clustering and self-organized agents applied to help network management, Expert Syst. Appl. 54 (2016) 29–47.
[8] G. Fernandes, J.J. Rodrigues, L.F. Carvalho, J.F. Al-Muhtadi, M.L. Proença, A comprehensive survey on network anomaly detection, Telecommun. Syst. 70 (3) (2019) 447–489.
[9] A. Löf, R. Nelson, Annotating network trace data for anomaly detection research, in: IEEE Conference on Local Computer Networks Workshops, 2014, pp. 679–684.
[10] C.S. Hood, C. Ji, Probabilistic network fault detection, in: IEEE Global Telecommunications Conference, Vol. 3, 1996, pp. 1872–1876.
[11] K. Papagiannaki, N. Taft, Z.-L. Zhang, C. Diot, Long-term forecasting of Internet backbone traffic, IEEE Trans. Neural Netw. 16 (5) (2005) 1110–1124.
[12] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, NetScope: Traffic engineering for IP networks, IEEE Netw. 14 (2) (2000) 11–19.

[13] R.-H. Hwang, M.-C. Peng, C.-W. Huang, P.-C. Lin, V.-L. Nguyen, An unsupervised deep learning model for early network traffic anomaly detection, IEEE Access 8 (2020) 30387–30399.
[14] P. Casas, F. Soro, J. Vanerio, G. Settanni, A. D'Alconzo, Network security and anomaly detection with Big-DAMA, a big data analytics framework, in: IEEE International Conference on Cloud Networking, 2017.
[15] T.-Y. Kim, S.-B. Cho, Web traffic anomaly detection using C-LSTM neural networks, Expert Syst. Appl. 106 (2018) 66–76.
[16] SAP Inc., SAP business suite, 2020, https://www.sap.com/products/business-one.html.
[17] A. Daneels, W. Salter, What is SCADA? in: International Conference on Accelerator and Large and Experimental Large Experimental Physics Control Physics Systems, 1999.
[18] V. Moreno, J. Ramos, J.L. Garcia-Dorado, I. Gonzalez, F.J. Gomez-Arribas, J. Aracil, Testing the capacity of off-the-shelf systems to store 10GbE traffic, IEEE Commun. Mag. 53 (9) (2015) 118–125.
[19] Lawrence Berkeley National Labs Network Research Group, The Libpcap library, 2022, https://www.tcpdump.org.
[20] V. Moreno, J. Ramos, P.M. Santiago del Río, J.L. García-Dorado, F.J. Gomez-Arribas, J. Aracil, Commodity packet capture engines: Tutorial, cookbook and applicability, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1364–1390.
[21] E. Miravalls-Sierra, D. Muelas, J. Ramos, J.E. López de Vergara, D. Morató, J. Aracil, Online detection of pathological TCP flows with retransmissions in high-speed networks, Comput. Commun. 127 (2018) 95–104.
[22] T. Zseby, E. Boschi, N. Brownlee, B. Claise, IP flow information export (IPFIX) applicability, 2009, RFC 5472.
[23] High Performance Computing and Networking research group, Dataset for workforce capacity planning for proactive troubleshooting in the network operations center, 2022, https://github.com/hpcn-uam/workforce-capacity.
[24] M. Mobilio, M. Orrù, O. Riganelli, A. Tundo, L. Mariani, Anomaly detection As-a-Service, in: IEEE Symposium on Software Reliability Engineering Workshops, 2019, pp. 193–199.
[25] Prometheus, How to use Prometheus for anomaly detection in GitLab, 2020, https://about.gitlab.com/blog/2019/07/23/anomaly-detection-using-prometheus/.
[26] R. Mijumbi, A. Asthana, M. Koivunen, F. Haiyong, Q. Zhu, Design, implementation, and evaluation of learning algorithms for dynamic real-time network monitoring, Int. J. Netw. Manage. 31 (4) (2021) e2108.
[27] D. Yao, X. Shu, L. Cheng, S.J. Stolfo, Anomaly detection as a service: challenges, advances, and opportunities, in: Synthesis Lectures on Information Security, Privacy, and Trust. Vol. 9, (3) Morgan & Claypool Publishers, 2017, pp. 1–173.
[28] J.D. Brutlag, Aberrant behavior detection in time series for network service monitoring, in: USENIX Conference on System Administration, 2000.
[29] G. Nguyen, S. Dlugolinsky, V. Tran, Á. García López, Deep learning for proactive network monitoring and security protection, IEEE Access 8 (2020) 19696–19716.
[30] G. Thatte, U. Mitra, J. Heidemann, Parametric methods for anomaly detection in aggregate traffic, IEEE/ACM Trans. Netw. 19 (2) (2010) 512–525.
[31] P. Barford, J. Kline, D. Plonka, A. Ron, A signal analysis of network traffic anomalies, in: ACM SIGCOMM Workshop on Internet Measurement, 2002, pp. 71–82.
[32] Y. Chen, K. Hwang, Collaborative change detection of DDoS attacks on community and ISP networks, in: IEEE International Symposium on Collaborative Technologies and Systems, 2006, pp. 401–410.
[33] B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, Sketch-based change detection: Methods, evaluation, and applications, in: ACM SIGCOMM Conference on Internet Measurement, 2003, pp. 234–247.
[34] R. Schweller, A. Gupta, E. Parsons, Y. Chen, Reversible sketches for efficient and accurate change detection over network data streams, in: ACM SIGCOMM Conference on Internet Measurement, 2004, pp. 207–212.
[35] D. Perdices, J.L. García-Dorado, J. Ramos, R. De Pool, J. Aracil, Towards the automatic and schedule-aware alerting of internetwork time series, IEEE Access 9 (2021) 61346–61358.
[36] I. Ucar, D. Morató, E. Magaña, M. Izal, Duplicate detection methodology for IP network traffic analysis, in: IEEE International Workshop on Measurements & Networking, 2013, pp. 161–166.
[37] D. Muelas, J.L. García-Dorado, S. Albandea, J.E. López de Vergara, J. Aracil, On the dynamics of valley times and its application to bulk-transfer scheduling, Comput. Commun. 164 (2020) 124–137.
[38] F. Mata, P. Żuraniewski, M. Mandjes, M. Mellia, Anomaly detection in diurnal data, Comput. Netw. 60 (2014) 187–200.
[39] Businesswire, Europe data center interconnect market - growth, trends, COVID-19 impact, and forecasts (2021 - 2026), 2021.
[40] J.L. Hintze, R.D. Nelson, Violin plots: A box plot-density trace synergism, Amer. Statist. 52 (2) (1998) 181–184.
[41] T. Favale, F. Soro, M. Trevisan, I. Drago, M. Mellia, Campus traffic and e-learning during COVID-19 pandemic, Comput. Netw. 176 (2020) 107290.
[42] P.A. Kullstam, Availability, MTBF and MTTR for repairable M out of N system, IEEE Trans. Reliab. R-30 (4) (1981) 393–394.
[43] P. Alavian, Y. Eun, K. Liu, S.M. Meerkov, L. Zhang, The (α, β)-precise estimates of MTBF and MTTR: Definition, calculation, and observation time, IEEE Trans. Autom. Sci. Eng. 18 (3) (2021) 1469–1477.
[44] J. Wu, M. N.B., J. Zhang, Flexible lognormal sum approximation method, in: IEEE Global Telecommunications Conference. Vol. 6, 2005, pp. 3413–3417.
[45] Grafana Labs, Beautiful metric & analytic dashboards, 2022, https://grafana.org.
[46] SLAC National Accelerator Laboratory, Network monitoring tools, 2022, https://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html#nmp.

Javier Ramos is an associate professor at Universidad Autónoma de Madrid (Spain). He received the M.Sc. degree in computer science and the Ph.D. degree in computer science and telecommunications from the Universidad Autónoma de Madrid, Spain, in 2008 and 2013, respectively. He was a visiting researcher at the Fraunhofer Institute for Open Communication Systems FOKUS, Germany (2012). His research interests are in the analysis of network traffic, quality of service, software defined networks and network function virtualization.

José Luis García-Dorado received his M.Sc. and Ph.D. degrees both in computer and telecommunications engineering from Universidad Autónoma de Madrid (UAM), Spain, in 2006 and 2010, respectively. He has been a member of the High Performance Computing and Networking (HPCN) research group at UAM since 2005. From then on, he was awarded a four-year predoctoral fellowship by the Ministry of Education of Spain (2007), and he was a Visiting Scholar with the Telecommunication Networks Group at Politecnico di Torino, Italy (2010), the Internet Systems Laboratory at Purdue University, USA (2013), and the Faculty of Applied Science at Universidad Técnica del Norte, Ecuador (2014 and 2015). Currently, he is an associate professor at UAM whose research interests are in the analysis of Internet traffic: its management, modeling, and evolution.

Javier Aracil received his M.Sc. and Ph.D. degrees (Honors) from Technical University of Madrid in 1993 and 1995, both in telecommunications engineering, and his five-year degree in mathematics from UNED in 2009. In 1995 he was awarded a Fulbright scholarship to pursue post-doctoral research at the University of California, Berkeley. In 1998 he was a research scholar at the Center for Advanced Telecommunications, Systems and Services of the University of Texas at Dallas. He was an associate professor for the University of Cantabria and Public University of Navarra. Currently, he is a full professor at UAM and a founding partner of the spin-off company Naudit HPCN. His research interests are in optical networks and performance evaluation of communication networks. He has authored more than 100 papers in international conferences and journals.