On the Modeling of Multi-Point RTT Passive Measurements for Network Delay Monitoring

Daniel Perdices, David Muelas, Iria Prieto, Luis de Pedro, Jorge E. López de Vergara

***Abstract*—Many network management actions need a simultaneous consideration of several elements' state. This is becoming an even more complex matter with the advent of reconfigurable deployments, where scaling functions up can prevent performance bottlenecks. Therefore, fine-grained detection of significant burdens arises as a cornerstone to optimize their monitoring and operation. We present AdPRISMA (Advanced distributed Passive Retrieval of Information, and Statistical Multi-point Analysis), a passive monitoring system intended to fit models for network delay measurements with clustering elements to improve representation of central and extreme behaviors. As distinguishing features, it relies on cost-effective multi-point round-trip time (RTT) passive network measurements, and is able to select a suitable parametric model optimizing the trade-off between fitting and complexity. AdPRISMA can correlate records collected from several vantage points and detect where performance issues are most likely to appear; adjust alarms in terms of the probability of events; and adapt its behavior to dynamic network conditions while presenting a fair identification of anomalous situations. We evaluate AdPRISMA with experiments both in virtual environments and with real-world data to provide evidences of its applicability and capabilities to represent network elements' delay.**

***Index Terms*—network monitoring, network delay, round-trip time, probability, passive measurements, performance management, pro-active management.**

I. Introduction

In recent times, network environments have turned from mostly static infrastructures to more flexible deployments, where software-based configurations are becoming common. With that, network managers have to tackle decisions that ground on simultaneously considering the state of several points of the network, to detect and solve possible performance burdens—*e.g.*, by scaling up affected network equipment in virtual networks, or by increasing the capacity of links. From the monitoring standpoint, methods and systems have to address these new possibilities and necessities of management activities, providing enhanced assistance for network operations [2].

Monitoring systems usually rely on active or passive measurements to detect possible issues. The use of the latter approach reduces risks in operational environments, as it provides Key Performance Indicators (KPIs) with minimal alteration of infrastructure. However, with the increasing heterogeneity of services and data rates in current deployments, passive data gathering is posing significant challenges. In this light, some recent network monitoring efforts are dealing with the thinning and capping of network data [3], and the exploitation of the distributed nature of these data to shift part of the analysis to the network equipment [4].

Moreover, network measurements must be robustly and consistently analyzed to opt for the most adequate decisions for incident solving and prevention. Thus, the application of suitable statistical modeling can improve pro-active policies, which motivates the application of methods that adapt to the evolution of KPIs [5]. This can help both to reduce false positive ratios and to automate actions, therefore simplifying management activities.

With these facts, we point to the following desirable characteristics for novel network monitoring solutions:

1) *Distributed and passive data gathering:* the retrieval of information should be distributed among different network elements. Monitoring systems should exploit capabilities of the equipment to improve scalability with a horizontal division of tasks. This can be implemented using several functionalities of common network equipment. For instance, we point to opportunistic retrieval from built-in capabilities (*e.g.*, exploitation of OpenFlow records); existing passive monitoring elements (*e.g.*, NetFlow or IPFIX exporters); and traffic forwarding based on SPAN ports or selective OpenFlow rules.
2) *Correlation of multi-point measurements:* measurements should be exploited to provide contextual data and link observations from different elements. As network issues usually affect complete segments, measurements that encompass only single points can hide the location, extension and nature of the problems. Therefore, correlation of measurements can provide deeper insights into performance issues and network state.
3) *Application of statistical models:* stochastic nature of network measurements requires a suitable statistical modeling. Otherwise, results may not reflect actual

Manuscript received March 18th, 2019; revised 6th June, 2019; accepted 15th June, 2019.

This paper is an extended version of the work presented in [1].

At the moment of writing this paper, all authors except Iria Prieto were with Universidad Autónoma de Madrid, Spain, and all authors except David Muelas were with Naudit High Performance Computing and Networking, S.L., Spain.

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under the project TRÁFICA (MINECO/FEDER TEC2015-69417-C2-1-R), by the European Commission under the project H2020 METRO-HAUL (Project ID: 761727), and by a collaboration scholarship of the Spanish Ministry of Education, Culture and Sports.

Cite as: D. Perdices, D. Muelas, I. Prieto, L. de Pedro, J. E. López de Vergara, "On the Modeling of Multi-Point RTT Passive Measurements for Network Delay Monitoring," *IEEE Transactions on Network and Service Management*, Volume 16, Issue 3, September 2019, pp. 1157-1169 DOI:10.1109/TNSM.2019.2924812

TABLE I: Parametric models included in AdPRISMA.

Parametric model	**Density function**	**Mode**
Normal(μ, σ)	$f(x\|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$	μ
Lognormal(μ, σ)	$f(x\|\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\{-\frac{(\ln x-\mu)^2}{2\sigma^2}\}$	$e^{\mu-\sigma^2}$
GEV(μ, σ, ξ)	$f(x\|\mu,\sigma,\xi) = \frac{1}{\sigma} t(x)^{\xi+1} \exp\{-t(x)\},\ t(x) = \begin{cases} (1+\xi(\frac{x-\mu}{\sigma}))^{-1/\xi} & \text{if } \xi \neq 0, \\ \exp\{-(x-\mu)/\sigma\} & \text{if } \xi = 0. \end{cases}$	$\begin{cases} \mu + \sigma\frac{(1+\xi)^{-\xi}-1}{\xi} & \text{if } \xi \neq 0, \\ \mu & \text{if } \xi = 0. \end{cases}$
Burr Type XII(α, c, k)	$f(x\|\alpha,c,k) = \frac{kc}{\alpha}\left(\frac{x}{\alpha}\right)^{c-1}\left(1+\left(\frac{x}{\alpha}\right)^{c}\right)^{-(k+1)},\ x,\alpha,c,k>0$	$\alpha\left(\frac{c-1}{kc+1}\right)^{\frac{1}{c}}$
α-stable(α, β, μ, σ)	No closed formula	No closed formula

network conditions and spurious values can lead to biased decisions. Models should consider a compromise between goodness of fit and complexity, to optimize analytics and prevent unnecessary computational costs.

4) *Robust data processing:* model fitting needs to include methods to extract relevant information from time-varying measurements. That fact entails a compromise between the granularity of detectable events and resiliency against noisy or isolated excursions.

In line with these trends, we present the design of Advanced distributed Passive Retrieval of Information, and Statistical Multi-point Analysis (AdPRISMA), an evolved version of our previous solution [1], intended to (i) passively collect network data (*e.g.*, traffic traces, flow records, ...) from several vantage points in the network; (ii) aggregate and filter data to estimate Round Trip Time (RTT) components corresponding to different network segments; and (iii) fit and select the most suitable statistical model for these measurements. We focus on RTT, as this specific KPI has been extensively used to detect and forecast network bottlenecks [6], [7].

Our proposal integrates several models that proved to fit RTT distributions fairly: AdPRISMA automatically ranks those models, and select the one with the highest goodness of fit and lowest complexity. Extending our previous work [1], this paper also shows how AdPRISMA can include data aggregation strategies, applying different time-windows and data projection methods. With this, it improves the trade-off between granularity of the events and resilience against artifacts or distortions.

Hereby, AdPRISMA constitutes a promising starting point to provide a flexible and general framework able to detect changes in the stochastic behavior of network KPIs while optimizing computational cost. We note that measures of centrality (*e.g.*, mean, median or mode) provide an easy-to-understand indicator of KPI departures. Remarkably, the mode is a significant value (*i.e.*, the most common value for a specific random variable), its estimation from a sample is challenging, and is robust against outliers and censored or truncated data [8].

The main contributions of our work are the study of RTT decomposition to facilitate the correlation of measurements and location of issues; and the definition of a methodology to rank models that puts together goodness of fit and complexity, paving the way for automated selection of the optimal statistical model for passive measurements. Additionally, we show that the statistical mode can be fairly obtained from the inferred models. The operation of AdPRISMA allows distributing the data collection process among several vantage points; correlating measurements retrieved from heterogeneous data sources; and it provides flexible models that adapt to changing behaviors. These aspects can help defining part of the system functionality in terms of OpenFlow rules, records exported to SDN controllers, and embed network monitoring functions in virtual networks, paving the way for improved monitoring processes in virtual and Software-Defined infrastructures.

To present our results, the rest of this paper is organized as follows: Section II reviews several related works that motivate our proposal. After that, Section III presents the architecture of AdPRISMA, describing the main functional blocks of our prototype, its operation and the method for the automation of model selection. On its part, Section IV assesses the functionality of the prototype, and reports the results of a case study that highlights the relevance of the model selection process. Finally, Section V discusses the findings of our study, and Section VI concludes the paper and depicts future work lines.

II. Related work

In this section, we present related works that motivate the design of AdPRISMA. We start with a review of statistical models for RTT measurements, to justify the selection of the models in our system. Then, we consider previous results that ground the assumption of validity of this representation, and methods to collapse individual flow estimates and obtain indicators of vantage points' performance. Finally, we focus on other monitoring frameworks that share design principles with our proposal.

A. Statistical models for RTT

Statistical modeling of network KPIs has deserved much attention, given its importance for network operation. This interest has resulted in a vast amount of literature reporting how different probability distributions represent network measurements, which extends to delay and RTT modeling. Table I compiles the parametric models included in our solution

(with closed expressions for density function and mode when available) to summarize the analysis of the literature.

Given their central position in inference, probability theory and empirical research [9], normal and lognormal models are a common approach when coping with data analysis. However, the research of KPIs in operational networks has exposed that many times they exhibit heavy-tailed behaviors in existing deployments, which grounded the exploration of more complex models able to capture large deviations [10]–[12]. As we will detail in the following sections, our system considers several parametric models (some of them with heavy tails) and compares their performance, taking into account different metrics to optimize the trade-off between goodness of fit and complexity. In [13], the authors explored which distribution adjusted single-hop delays in computer networks. Their conclusions pointed to a good representation of this KPI with Weibull distributions, as delays presented fair unimodal behaviors. Similar results were reported in [14], while in this latter case multi-modal behaviors were observed (somehow expectable, as that work analyzed end-to-end delays) so mixtures of Weibull distributions provided good fitting to the measurements. Inspired by these results, we explored two additional parametric families, which for some values in the space of parameters lay near Weibull distributions.

On the one hand, we have considered the Generalized Extreme Value (GEV) distribution [15], given their suitability to represent variables with large and rare values. Remarkably, GEV distributions generalize Weibull, Gumbel and Fréchet distributions, which motivates the selection of this model. On the other hand, we also introduced Burr Type XII distributions to model RTT, motivated by the relation of this parametric family with Weibull distributions [16]. The complexity of both models is comparable to Weibull distributions, but their broader flexibility can potentially reduce deviant cases.

Additionally, in recent times α-stable distributions have been applied to model RTT [12]. This family is very flexible and general, but much more complex that those previously commented. In fact, the fitting of the parameters of α-stable distributions is computationally expensive [17], [18] and there is no closed expression for their density function. Remarkably, α-stable distributions appear in the generalized central limit theorem and converge to normal distributions for some values in the space of parameters.

B. Time-varying network modeling

As stated above, the properties of several parametric distributions offer a promising framework to describe RTT and delay components in operational networks. While this is a primary step along data modeling, model fitting requires a time-dependent consideration to guarantee that such representation can exist. In other words, we wonder to what extent measurements are stable—i.e., they follow a given distributional law during observations periods.

RTT can be decomposed, as we develop below, in delay components that affect network traffic along a path. Therefore, a reasonable condition for RTT stability is the stability of those components. Given the importance of such components in overall network performance, several previous works have addressed its modeling and understanding. For instance, a formal model for stochastic components of delay in common network equipment is presented in [19]. In that work, the authors pointed to relevant factors—remarkably, network load and node capacity—that provided a suitable estimator for delay components.

In that same line, we consider that under stationary network load and in the absence of changes in node capacity, delay components can be considered *short-term invariants*: this seems a reasonable condition, as a result of empirical and grounded analysis of network load [20]. In this latter work, authors analyzed a method for the detection of abrupt short-time changes in network load, showing that under very general assumptions this indicator can be considered invariant. Then, they applied cluster aggregation to test the model compliance—specifically, multivariate Gaussian model—and detected excursions from the typical behavior.

Following a somehow similar approach, we put together these previous results to define a projection method to prefilter individual flow estimations in time windows, selecting representatives for central and extreme values. Then, we characterize the typical behavior of projections using the aforementioned models. Additionally, we also introduce a control measurement to assure stability of the estimates using Dictyogram [21]. This method describes the evolution of flow characteristics by accounting the frequency of their values within a set of order statistics. Hence, it provides a flexible evaluation of changes in the distribution with the analysis of its corresponding variation rates.

C. Multi-point distributed monitoring systems

Beyond improving techniques to retrieve information from measurements, network monitoring and analysis solutions need to suit novel operational architectures. This entails that data capture and deployment processes should evolve towards more scalable and flexible approaches. As an illustrative situation, current trends regarding network slicing and virtual networks on top of shared hardware require this type of approaches to gather data without incurring in high costs—*e.g.*, movement of big data volumes. This matter is not a particularly new concern for network monitoring, and many previous results explored principles that can help to improve current systems.

For instance, the design of cooperative monitoring systems [22] arose as a promising approach to alleviate the shortcomings of monitoring scalability. These classical ideas can pave the way for improved solutions in the network monitoring scope, as stated in [2]. The architecture of AdPRISMA shares many of the principles that guided these proposals.

Even more important is that many current network monitoring efforts are focused on how to take advantage of the ever increasing capabilities of network equipment. This opens the gate to disaggregate network monitoring, moving specific tasks to the most suitable equipment in the network. Turboflow [4] is a recent proposal that relies on the embedding of flow generation into programmable switches. However, the

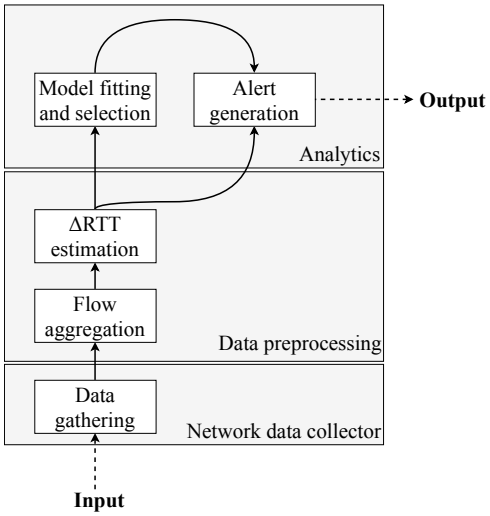


Fig. 1: Functional modules of AdPRISMA.

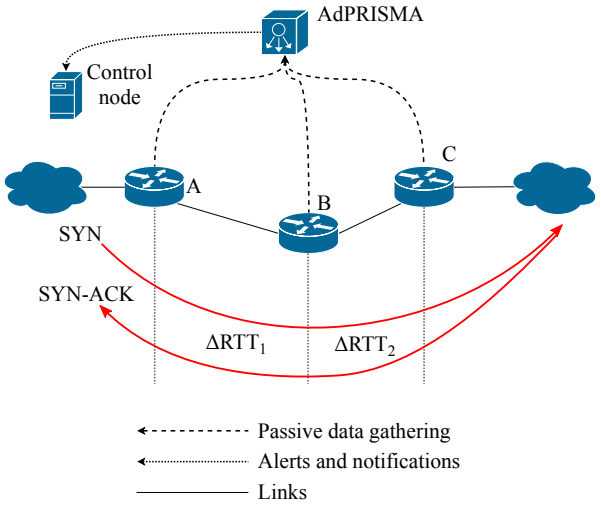


Fig. 2: Operation of AdPRISMA. Red arrows represent a sample connection traversing the three monitored points of the network, distinguishing the different RTT components that are estimated to detect possible bottlenecks.

authors of that work highlight that stateful information may limit the complete implementation of some processes in the network hardware. In the same line, Sonata [23] distributes monitoring tasks to different network elements, providing a query-based API that can be exploited by other modules. Parallel to these proposals, AdPRISMA provides high-level analytics after aggregation and correlation of traffic packets or flow records that may be produced by different sources and methodologies.

Finally, and regarding the trends in virtualization and software-definition of networks, we point to other recent works that exploit containers to define flexible monitoring services that can be instantiated on demand and linked to specific applications [24]. The modular design of AdPRISMA is totally aligned with these trends, providing a higher decoupling of data gathering and analytics. Such approaches can push network monitoring proposals toward microservice-oriented architectures [25].

III. System Architecture and Description

Along this section, we describe the main functional components of AdPRISMA, which are summarized in Fig. 1. In the current proof of concept implementation, AdPRISMA relies on flow records to conduct the analysis and modeling of RTT. To prevent ambiguities, we clarify that hereafter we refer to *TCP flow* as a set of TCP packets with a common 4-tuple, which traverse a particular vantage point in the network during a specific time interval, as stated in RFC 7011 [26].

Additionally, we synthesize the operation of AdPRISMA in Fig. 2. First of all, passive measurements are gathered from the available vantage points. These measurements are aggregated in AdPRISMA, and correlated to obtain estimations of RTT and its components—that is, the increments along the network segments defined by vantage points. After that, the system fits and selects the parametric model for measurements, and provides estimations of significant central values—*e.g.*, mean, median and mode— and other order statistics such as extreme values. This leads to flexible and adaptable profiles for alerts, thus providing indicators of performance issues.

In the following, we detail these operations and how they are implemented within the different functional blocks. For our description, we follow a constructive approach that first considers how data are gathered and preprocessed, and then details how they are exploited to build the models.

A. Data gathering and preprocessing

Flows are collected in several ways. Some examples are Netflow or IPFIX [26], and other custom tools that send at least information about when every flow starts. Except for special cases, these timestamps are taken from SYN and SYN-ACK segments, which let us have an estimation of RTT that only requires that both flows of the same connection are sampled. Regarding performance issues in this process, we may distinguish two different situations: flow aggregation in a computing element different to network equipment, and aggregation inside the networking elements. In the first case, it is possible to capture traffic up to 40 Gb/s in commodity hardware—e.g., see [27], [28]. In the case of monitoring functions within network equipment, performance issues may appear depending on traffic characteristics and capabilities of specific hardware, while commercial equipment includes support for these operations.

AdPRISMA estimates RTT by subtracting the start times of two TCP flows that share temporal and spatial localities, and the 4-tuple swapping source and destination addresses and ports. Then, it correlates *equivalent flows*: TCP flows sharing the 4-tuple and time interval but observed in different points of presence. This process is described in Algorithm 1.

Once RTT is estimated and correlated, the *equivalent flow* contains information of the flow in several locations. By looking at Fig. 2, we observe that RTT in hop j is given in (1):

$$\mathrm{RTT}_j = \sum_{i=j}^{N} \Delta\mathrm{RTT}_i \tag{1}$$

Algorithm 1 Flow aggregation.

```
function getSuperflows(flows...)
table ← InitializeSuperFlowsTable()
for flow in flows do
    if flow is ip and tcp then
        if flow.srcPort < flow.dstPort or
        (flow.srcPort = flow.dstPort and
        (flow.srcIp < flow.dstIp)) then
            quintuple ← (flow.srcIp, flow.srcPort, flow.dstIp,
            flow.dstPort, flow.ipProto)
        else
            quintuple ← (flow.dstIP, flow.dstPort, flow.srcIp,
            flow.srcPort, flow.ipProto)
        end if
        table[quintuple].addFlow(flow)
    end if
end for
return(table)
```

By inverting this linear operator, we obtain an estimation of the component in the network segment between vantage point j and $j+1$ as in (2):

$$\Delta\mathrm{RTT}_j = \mathrm{RTT}_j - \mathrm{RTT}_{j+1} \tag{2}$$

Note that, in contrast to one-way delay measurements, these estimations do not require clock synchronization, since the RTT_j are absolute values, each one computed with the flow estimates performed in the same vantage point j with its local clock as a single reference time. As this process involves a single clock source, synchronization among different vantage points is not required.

B. Model selection and adaptation

Due to the stochastic nature of network measurements, statistical models are needed. In our case, these models are intended to characterize $\Delta\mathrm{RTT}_j$ behavior, so that frequent events can be distinguished from anomalies or deviant events.

Apart from how challenging model fitting can result, the selection of an optimal model to be used emerges as key matter for systems as ours. For this aim, we have equipped AdPRISMA with several criteria, summarized in Table II, to adapt its behavior to a wide range of situations:

1) *Coefficient of Determination ($\mathbf{R}^2$)*: A well-known metric of goodness of fit is the coefficient of determination, $\mathbf{R}^2$. This metric is based on a linear fitting of (x_k, y_k), where x_k are the order statistics of the sample and y_k are the corresponding quantiles of the model. If the samples follow the model, there must be a strong linear relation, which entails that $\mathbf{R}^2$ must be close to 1. This is a necessary but no sufficient condition [29], so although this method cannot provide a formal proof of goodness of fit, it can be applied to rule out the parametric models with the lowest values—*i.e.*, to select that with the strongest linear relation between the order statistics of the sample and estimated distribution.

TABLE II: Summary of metrics for model selection.

Metric	Description	Expression
$\mathbf{R}^2$	Only considers fitting.	$1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}}$
AIC	Considers both fitting and number of parameters.	$2(k - \log(\hat{L}))$
BIC	Considers fitting, number of parameters and sample size.	$\log(N)k - 2\log(\hat{L}))$

2) *Akaike Information Criterion (**AIC**)*: This a statistical method to compare different models based on two factors: complexity and goodness of fit. It has the expression in (3):

$$AIC = 2(k - \log(\hat{L})) \tag{3}$$

where k is the number of parameters of the model and $\hat{L}$ is the maximum of the likelihood function [30]. It is remarkable that complexity is just evaluated with the number of parameters, and this makes it an optimistic approach.

3) *Bayesian Information Criterion (**BIC**)*: Related to the aforementioned AIC, it introduces an additional component, which is the number of samples. This is intended to reduce overfitting in parametric models, so that the complexity and goodness of fit are balanced [31]. It is defined as in (4):

$$BIC = \log(N)k - 2\log(\hat{L}) \tag{4}$$

where N stands for the sample size and the rest of variables were described in AIC.

These three criteria allow choosing the most appropriate model based on complexity and goodness of fit, and on the situation and requirements of the other top-level system that use this information. For instance, for real-time applications, simpler models are preferred so the model computation is not a bottleneck in the monitoring system.

Finally, and taking into account previous discussion about time-varying KPI distributions, we note that these criteria provide a basis for adaptive systems. That is, if none of the goodness of fit metrics above point at a suitable model, observations could be partitioned in subsets to represent behavior with several distributions.

C. Aggregation of single-flow estimates

Given the high variance of single-flow-based estimates, we envisaged an aggregating procedure to better characterize vantage point modeling. In other words, as AdPRISMA is intended to provide indicators for issues at network elements or segments, spurious variance in flow behavior could lead to biased conclusions. To separate this latter type of deviant situations from sustained changes in the vantage points' behavior, we have introduced a windowed filtering of single-flow observations.

We recall that AdPRISMA tries to obtain a model for the distribution of $\{\Delta\mathrm{RTT}_i\}$ for separate network segments. Therefore, it requires some stability on the fitted distribution. In this regard, we detected two main issues that may appear because of the flows stochastic behavior. On the one hand,

changes on the underlying distribution can lead to sub-optimal adjustments—e.g., a sustained change on the expectation of $\{\Delta\text{RTT}_i\}$. On the other hand, the convergence to a fair approximate of the distribution depends on the number of observations, as stated in the Glivenko-Cantelli theorem [32].

This entails a bias-variance trade-off—i.e., the balance between how far is the estimated model to the theoretical distribution, and how the model fits the observations. AdPRISMA copes with this matter by clustering single-flow observations in time windows. We distinguished two possible strategies to proceed with this aggregation.

On the one hand, the first strategy is intended to extract a *central* representative within each time window. This would lead to robust models for typical behaviors sustained along time. This can be accomplished using a projection as in (5):

$$\Delta\text{RTT}(t) = \arg\min_{x\in\mathbb{R}} \sum_{j\in\mathbb{J}_t} (d(\Delta\text{RTT}_j, x)),\ t \in \mathbb{T} \tag{5}$$

where $t \in \mathbb{T}$ represents the time-domain partition, $\mathbb{J}_t$ the index set of $\{\Delta\text{RTT}_i\}$ within each element of the time-domain partition, and $d(\cdot,\cdot)$ a distance—e.g., any L^p distance. The projection can lead to different *centroids*, such as the median (L^1) or average (L^2) for the observations within the interval. With this, AdPRISMA introduces a variance-reduction procedure that can attenuate the effect of flow distortions—i.e., isolated extreme values do not affect model fitting.

On the other hand, the second strategy pursues the determination of boundaries for extreme values. This is accomplished **by** using order statistics of the observations within each time window as in (6):

$$\Delta\text{RTT}_p(t) = \inf\left\{x : p \le \frac{1}{|\mathbb{J}_t|}\sum_{j\in\mathbb{J}_t} \mathbb{1}_{[0,x]}(\Delta\text{RTT}_j)\right\},\ t \in \mathbb{T} \tag{6}$$

where $t \in \mathbb{T}$ represents the time-domain partition, $\mathbb{J}_t$ the index set of $\{\Delta\text{RTT}_i\}$ within each element of the time-domain partition, $\mathbb{1}_A(t)$ the indicator function of set A—i.e. its value is 1 if $t \in A$ and 0 otherwise—and $p \in [0,1]$ indicates the selected probability level. This approach is useful to define and model extreme values' bounds with sensitivity to trends along time.

Bias control on projections is accomplished with the study of convergence to a robust empirical estimate of the theoretical distribution function. Dictyogram [21] offers a basis for quantitative criteria to determine whether the time-domain partition suffices to a reasonable convergence—i.e., if the number of observations offer a fair representation of the distribution. Dictyogram maps time-depending study of the distribution of a flows' characteristic, such as the ΔRTT, onto the analysis of the number of flows lying on categories defined in terms of order statistics of the specific characteristic.

To do so, once the values corresponding to a grid of probability levels $\{x_k\}_{k=1,\dots,N}$ are selected, then flows can be partitioned using the intervals in (7):

$$\mathcal{I}_k = \begin{cases} [0, x_1] & \text{if } k = 1 \\ (x_{k-1}, x_k] & \text{if } 1 < k < N+1 \\ (x_N, \infty) & \text{if } k = N+1 \end{cases} \tag{7}$$

These intervals induce a set of time series with the number of flows within each interval and time window, which we denote as $f_k(t)$ in (8):

$$f_k(t) = \sum_{j\in\mathbb{J}_t} \mathbb{1}_{\mathcal{I}_k}(\Delta\text{RTT}_j) \tag{8}$$

Using these series, we can define a relative measure of variation along time as presented in (9):

$$d[f_k(t)] = \frac{\sum_k |f_k(t) - f_k(t-1)|}{\sum_k f_k(t)},\ t \in \mathbb{T} \tag{9}$$

which accounts for the cumulative relative variation of the number of flows in each category.

Then, bursts in $d[f_k(t)]$ are equivalent to abrupt variations in the Empirical Cumulative Distribution Functions (ECDFs) in adjacent time windows. In other words, the stability of this function offers a quantification of ECDFs' stability.

D. Summarizing the models: centrality measures

Once a suitable model—any for central or extreme values—has been fitted to data, computation of some relevant statistical summaries can be performed. In this light, the mode of a sample is a prominent centrality measure that returns the most probable value of a distribution. However, and given that finding a good parametric model is not always feasible, we also evaluated alternative methods to estimate the mode.

We have considered methods for the univariate case—see the analysis in the introduction of [8]—and studied both indirect (that is, relying on a non-parametric density function estimation) and direct (essentially, search methods around intervals where the mode is likely to appear) proposals:

1) Estimation through the Kernel Density Estimator (KDE): This approach arises from the definition of mode. First, the KDE, a PDF estimator, is calculated. The mode is estimated as the maximum of the KDE, as in (10):

$$\widehat{\text{Mode}}(X) = \arg\max_{x\in\mathbb{R}} \hat{f}(x) \tag{10}$$

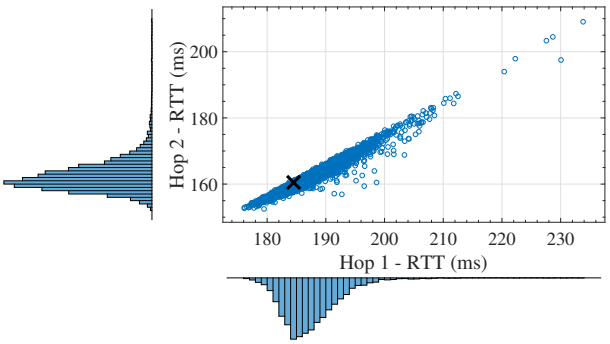
While this method can reveal important details about the density function (*e.g.*, shape or number of modes), it depends on the convergence of KDE to the actual PDF.

2) Half-Sample Mode (HSM) algorithm: The HSM algorithm is a robust and fast method to approximate the mode [33]. This algorithm is based on the principle that "the mode is in the smaller interval that contains half of the sample". By applying this idea, we reduce both computations and assumptions, making this approach a good one to use in many situations.

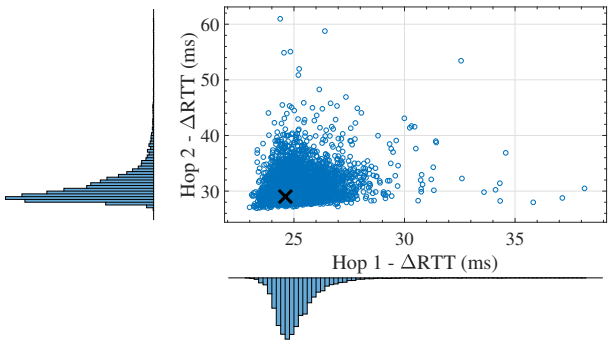
IV. Evaluation

A. Experimental design

The validation of AdPRISMA proof of concept encompassed three different stages: first one, with laboratory experiments, where we tested the system in controlled environments; second and third ones with real data coming from different enterprise data centers.



(a) Scatter plot of hops 1 and 2 for RTT.



(b) Scatter plot of hops 1 and 2 for ΔRTT.

Fig. 3: Results for the virtual environment. The × shows the intersection of the modes of hops 1 and 2.

TABLE III: Estimated mode of ΔRTT_1 and ΔRTT_2 in the virtual environment, for each of the methods.

	ΔRTT_1				ΔRTT_2			
Model	**Mode**	**R^2**	**AIC**	**BIC**	**Mode**	**R^2**	**AIC**	**BIC**
KDE	24.669ms	-	-	-	28.817ms	-	-	-
HSM	24.699ms	-	-	-	28.755ms	-	-	-
Normal	25.125ms	0.811	-54393.423	-54380.389	30.561ms	0.789	-44919.993	-44906.959
Lognormal	24.665ms	0.827	-54897.417	-54884.382	28.893ms	0.830	-45842.766	-45829.732
GEV	24.640ms	0.948	**-56638.769**	-56615.216	28.951ms	0.976	-48284.294	-48264.742
Burr Type XII	24.650ms	0.956	-56636.163	**-56616.612**	28.657ms	**0.996**	**-48324.508**	**-48304.956**
α-stable	~24.650ms	**0.970**	56625.412	-56599.344	~29.098ms	0.272	-48052.870	-48026.801

The first stage was accomplished in an emulation-based experimental environment[1] on top of mininet [34], [35]. Virtual networks were deployed in commodity hardware (a laptop PC with a quad-core processor, 8GB RAM) and configured as follows:

1) Create a linear topology with either routers or switches as non-terminal nodes. Specifically, we used 6 hops in our experiments.
2) Use `netem` and `tc` in each link to establish a delay of $(10 + 2i)$ ms, where i is the index of the hop, and a capacity of 20 Mbit/s.
3) Capture traffic passing through each interface. As our method only needs TCP packets with the SYN flag activated, this capture did not exert a significant impact on the performance of the environment.
4) Configure terminal nodes as traffic generators. We used these nodes to generate TCP connections that go through all hops.

In order to make this situation closer to a real network, background traffic is introduced. Several techniques were used to generate such load: (i) ICMP ping with random intervals, (ii) `iperf` and (iii) traffic generators that rely on TCP connections to a conventional TCP or HTTP server [36].

After validation, there were no significant divergences between the measurements when using any of these methods. Therefore, we configured several nodes to send ICMP packets of size 1000 Bytes at random intervals in bursts of 500-1000 packets to simplify the experiments.

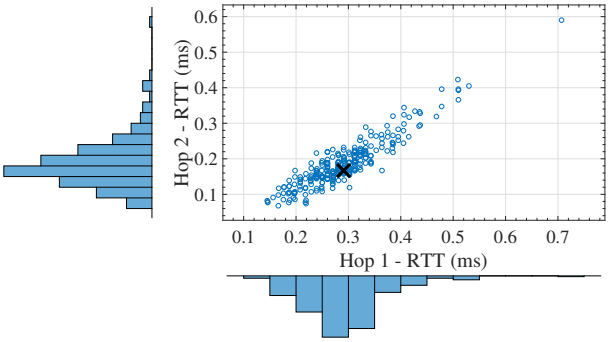
In the second stage of experiments, we analyzed flow records from a data center network with AdPRISMA to assess its outcomes in an actual case study. This dataset, hereinafter denoted *Dataset*$_1$, has the following characteristics:

- It includes real traffic traces of core and service switches, load balancers and virtual machines in operation, gathered from an Internet Service Provider (ISP) data center network.
- It was captured using the management software of two vantage points, so no special equipment was completely dedicated to network monitoring.

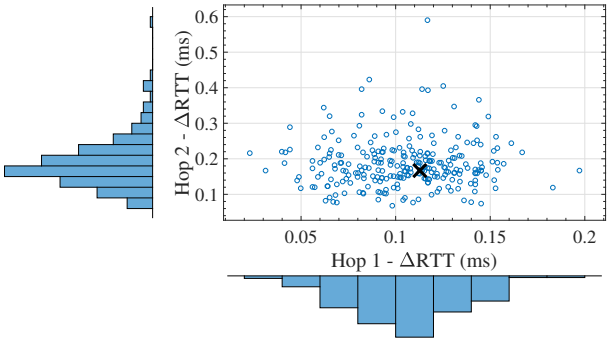
Due to the presence of some outliers in the second hop of the dataset, some preprocessing was applied to visualize and plot the data. As some of the destinations of the connections are virtual machines, the outliers were likely caused by the hypervisors managing virtual machines.

Finally, the third experimental stage was intended to assess the validity of fixed distribution models along longer observation periods. For this purpose, we consider a second real dataset (*Dataset*$_2$) retrieved from an enterprise datacenter, not analyzed in our previous work [1]. In contrast to *Dataset*$_1$, *Dataset*$_2$ is composed of roughly seven working hours of traffic captured in two vantage points deployed to monitor the performance of an in-between operational firewall during working hours in one day. This network presents heavy traffic load during the peak hour around 9:00 AM, and some other minor peak moments during the rest of the day. Whereas *Dataset*$_1$ lasted for only some few minutes, this latter case represents a scenario where single-flow estimates might not be stable along time. With this, we assessed the principles that grounded the projection methodology to obtain node indicators from individual flow estimates—trying to capture a fair representation of the global node performance.

[1]The source code is available at https://github.com/hpcn-uam/mininetplus .



(a) Scatter plot of hops 1 and 2 for RTT.



(b) Scatter plot of hops 1 and 2 for ΔRTT.

Fig. 4: Results for $Dataset_1$. The × shows the intersection of the modes of hops 1 and 2.

TABLE IV: Estimated mode of ΔRTT_1 and ΔRTT_2 in $Dataset_1$, for each of the methods.

	ΔRTT_1				ΔRTT_2			
Model	**Mode**	$\mathbf{R^2}$	**AIC**	**BIC**	**Mode**	$\mathbf{R^2}$	**AIC**	**BIC**
KDE	0.1080ms	-	-	-	0.161ms	-	-	-
HSM	0.1124ms	-	-	-	0.167ms	-	-	-
Normal	0.1042ms	0.990	**-4754.719**	**-4747.567**	0.7553ms	0.05	-1728.580	-1721.429
Lognormal	0.0861ms	0.905	-4711.002	-4703.850	0.1223ms	0.295	-4093.103	-4085.950
GEV	0.1055ms	0.992	-4750.049	-4739.321	0.1251ms	0.651	-4251.231	-4240.504
Burr Type XII	0.1063ms	**0.995**	-4754.618	-4743.890	0.1546ms	0.708	-4275.132	-4264.402
α-stable	~0.1042ms	0.991	-4750.721	-4282.579	~0.1568ms	**0.970**	**-4296.883**	**-4282.579**

B. Results in virtual environment

We recall that the delay among nodes was controlled by `netem` in the virtual environment, so the double of the configured delay is expected as theoretical ΔRTT. The effect of traffic load increases somehow this bound, and thus estimated ΔRTT should be slightly higher.

Fig. 3 shows scatter plots of the RTT and ΔRTT in the two first hops—there were no significant differences with measurements in the other vantage points, so we omit the consideration of every combination for the sake of brevity. After truncating extreme values to improve visualization, the scatter plot for the latter shows a concentrated set of points around the mode with skewed density functions. Additionally, Table III summarizes the results of modeling in AdPRISMA.

The results show that GEV, Burr Type XII and α-stable distributions are close enough ($\mathbf{R^2} > 0.90$) to be considered fair models. Table III illustrates the value of the multi-metric ranking. According to the $\mathbf{R^2}$, the preferred model would the sophisticated α-stable distribution. However, **AIC** points to GEV as optimal model, because its lower complexity (respect to α-stable distribution) compensates the loss in goodness of fit. Finally, **BIC** considers the Burr Type XII as the best model. These results show how AdPRISMA can be tuned to take into account and balance several factors (complexity, number of samples or just goodness-of-fit) depending on the context.

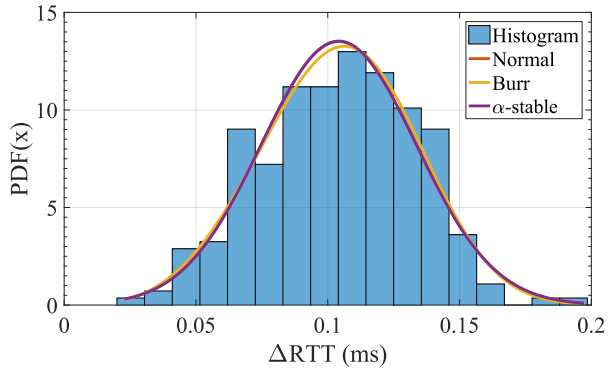
Table III also includes estimated modes, including the computations with KDE and HSM. In each case, we observe than the mode is around the theoretical expected values, both for ΔRTT_1 and ΔRTT_2 (24 and 28ms, respectively) plus an additional delay of ~0.7 ms because of the background traffic. In this case, it is worth noting that the mean (mode estimator for the normal model) suffered from variable deviations with respect the expected value, depending on the skewness of the ΔRTT distribution. In the case of ΔRTT_2, α-stable model also exhibited high distortions, due to numerical errors during parameter estimations.

C. Analysis of a data center network

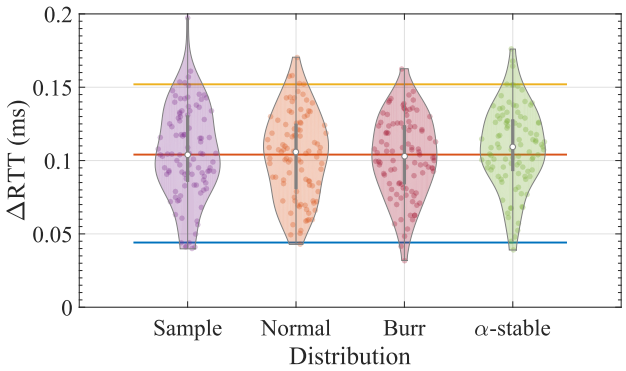
Once we have assessed the accuracy of AdPRISMA, we inspect the results obtained during the study of a real data center network. In a similar way to the previous experiments, we present scatter plots of RTT and ΔRTT in Fig. 4, and summarize the results of model fitting and mode estimation in Table IV. Additionally, we include in Fig. 5 the representation of sample data compared to the three models that provided the best goodness of fit. Fig. 5a and 5c present the comparison among the estimated densities and the normalized sample histogram, and Figures 5b and 5d depict the corresponding violin plots with some remarkable order statistics—specifically, the median as centrality measure, and the 5^{th} and 95^{th} percentiles for extreme values.

For ΔRTT_1 (*i.e.*, measurements in the first vantage point), Burr Type XII model obtained the highest $\mathbf{R^2}$, whereas **AIC** and **BIC** suggest that a normal model is also reasonable and much less complex. This behavior is coherent with the insights from Fig. 5b, where Burr Type XII presents higher accordance with the order statistics of the sample, while the adjusted normal model fairly fits the sample distribution.

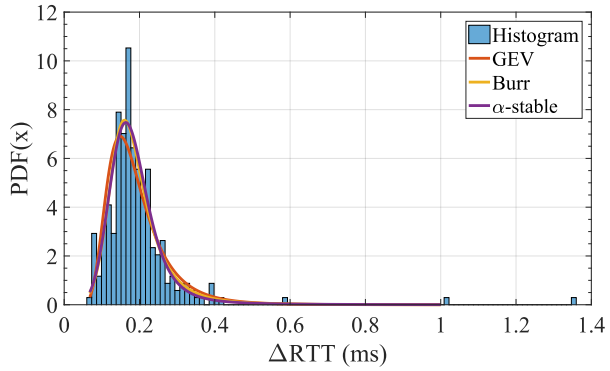
However, the behavior of ΔRTT_2 (*i.e.*, measurements in the second vantage point) is very different. In this case, the preferred model is the α-stable distribution, with better scores (either when considering $\mathbf{R^2}$, **AIC** or **BIC**) for any other option. The skewness and tail of ΔRTT_2 prevent from considering more simplistic models, with poor accuracy in the



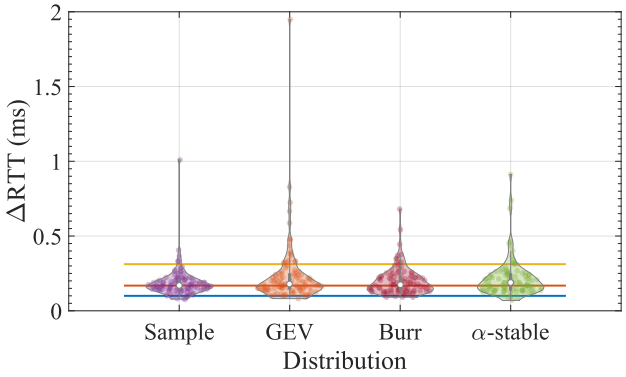
(a) Histogram and density comparison, ΔRTT_1.



(b) Violin plots, ΔRTT_1. Horizontal lines show percentiles 5th, 50th and 95th of the sample.



(c) Histogram and density comparison, ΔRTT_2.



(d) Violin plots, ΔRTT_2. Horizontal lines show percentiles 5th, 50th and 95th of the sample.

Fig. 5: Comparison among models and observation for ΔRTT_1 and ΔRTT_2 in *Dataset*$_1$.

representation of the shape and order statistics of the sample distribution—see Figures 5c and 5d for illustration.

This situation exposes two important matters. First, this dataset provides evidences of disparity in the behavior of RTT components among vantage points. That is, we cannot assume the existence of a one-fits-all model for network KPIs, even within the same network. Moreover, our results show that complex models with outstanding performance in some situations can fail where simpler ones achieve good results. Additionally, this analysis shows that RTT components (i.e., ΔRTT) locate and differentiate how traffic is affected when traversing each of the vantage points. This fact is useful to detect situations of local saturation in a network segment that are not detectable with the aggregated RTT.

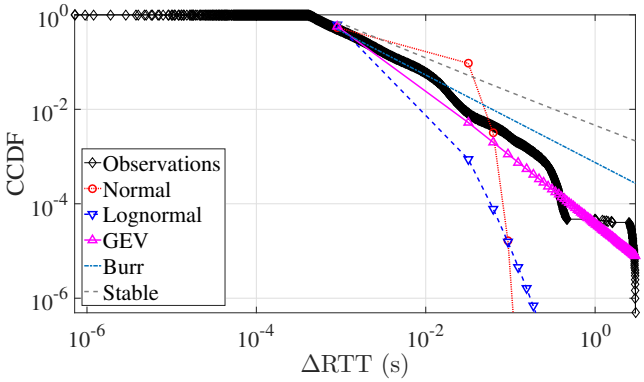


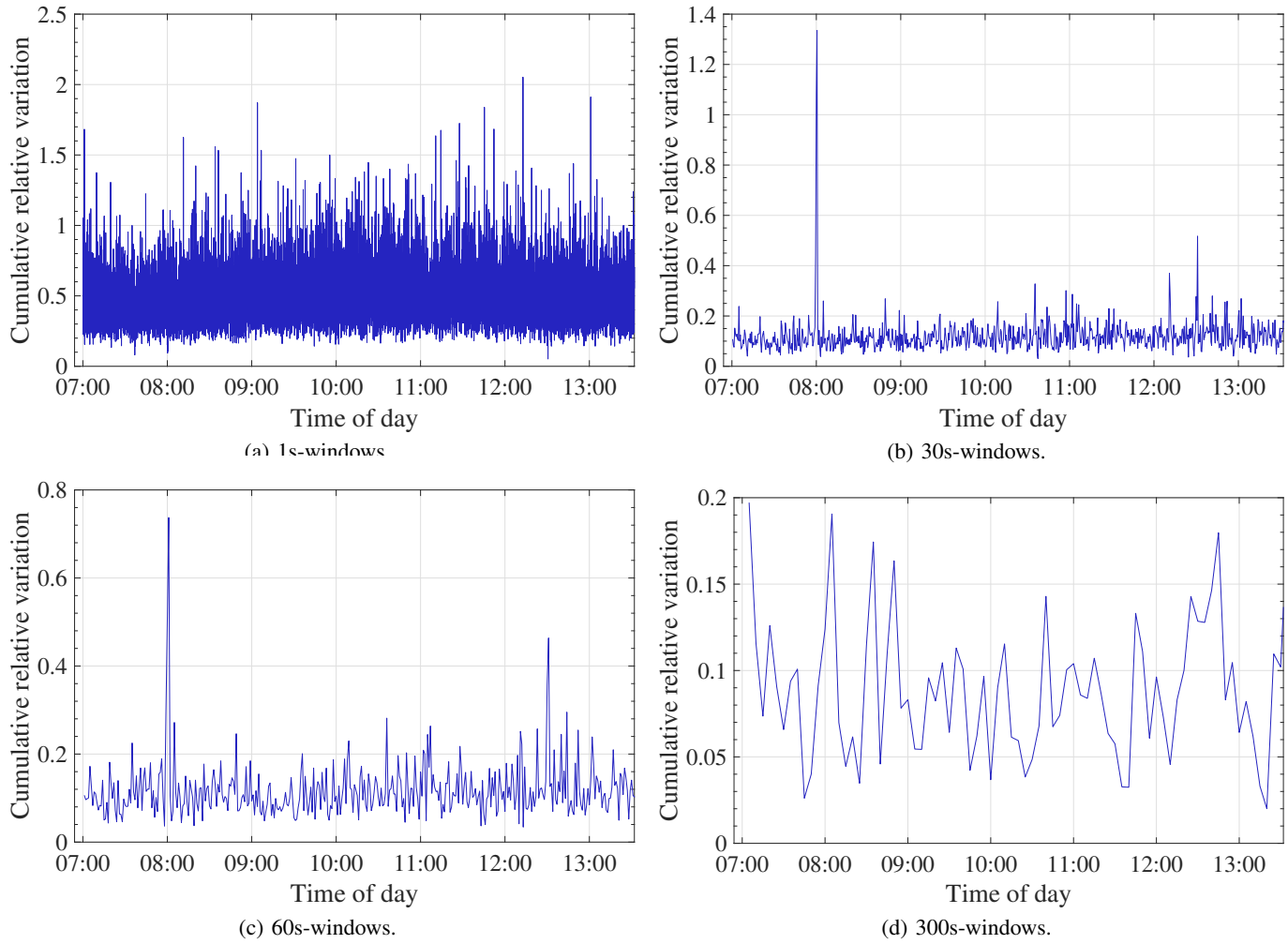
Fig. 6: ΔRTT distribution, single-flow estimates, *Dataset*$_2$.

D. Analysis of network equipment

Once the short data ranges' modeling have provided good results for the characterization of real measurements, we move forward to the evaluation of its outcomes in longer periods. To do so, we considered the ΔRTT extracted from *Dataset*$_2$, which lasts for several hours and exhibits severe extreme ΔRTT values as a result of the firewall operation.

In this case, the situation is completely different due to the bursty nature of the single-flow estimates. Fig. 6 shows this situation with the Complementary Cumulative Distribution Function (CCDF) of ΔRTT for all the flows in the trace, showing that the aforementioned models cannot fit either central or extreme values. In this scenario, projections within different time windows—1s, 30s, 60s, 300s—can reduce the variance in centrality measures caused by isolated extreme values.

As stated before, Dictyogram enables the definition of quantitative metrics to evaluate ECDFs' stability along time and determine whether a window size may be suitable. Fig. 7 shows the cumulative relative variation of Dictyogram for the aforementioned window sizes, by applying (9). Remarkably, the lower the window size is, the projection effect will become



(a) 1s-windows.
(b) 30s-windows.
(c) 60s-windows.
(d) 300s-windows.

Fig. 7: Dictyogram cumulative relative variation in $Dataset_2$. Effect of window size for computation of flows per decile.

less noticeable—i.e. models of projected observations can be expected to be similar to the single-flow ones.

Hence, robust fitting of central delay values will require larger windows sizes although either excessively coarse or fine-grained pre-filtering can obfuscate significant punctual deviations—see the peaks at 8:00 and 12:30 in Figures 7b and 7c, which are undetectable in Figures 7a and 7d.

These situations clearly translate into different fitted models after projection, as shown in Fig. 8. This figure illustrates that the coarser the projection is, the better the model fits: both Figures 8a and 8b present cases where few extreme observations impoverish the fitting, whereas Figures 8c and 8d display models that fairly fit the data up to 99$^{\text{th}}$ percentile.

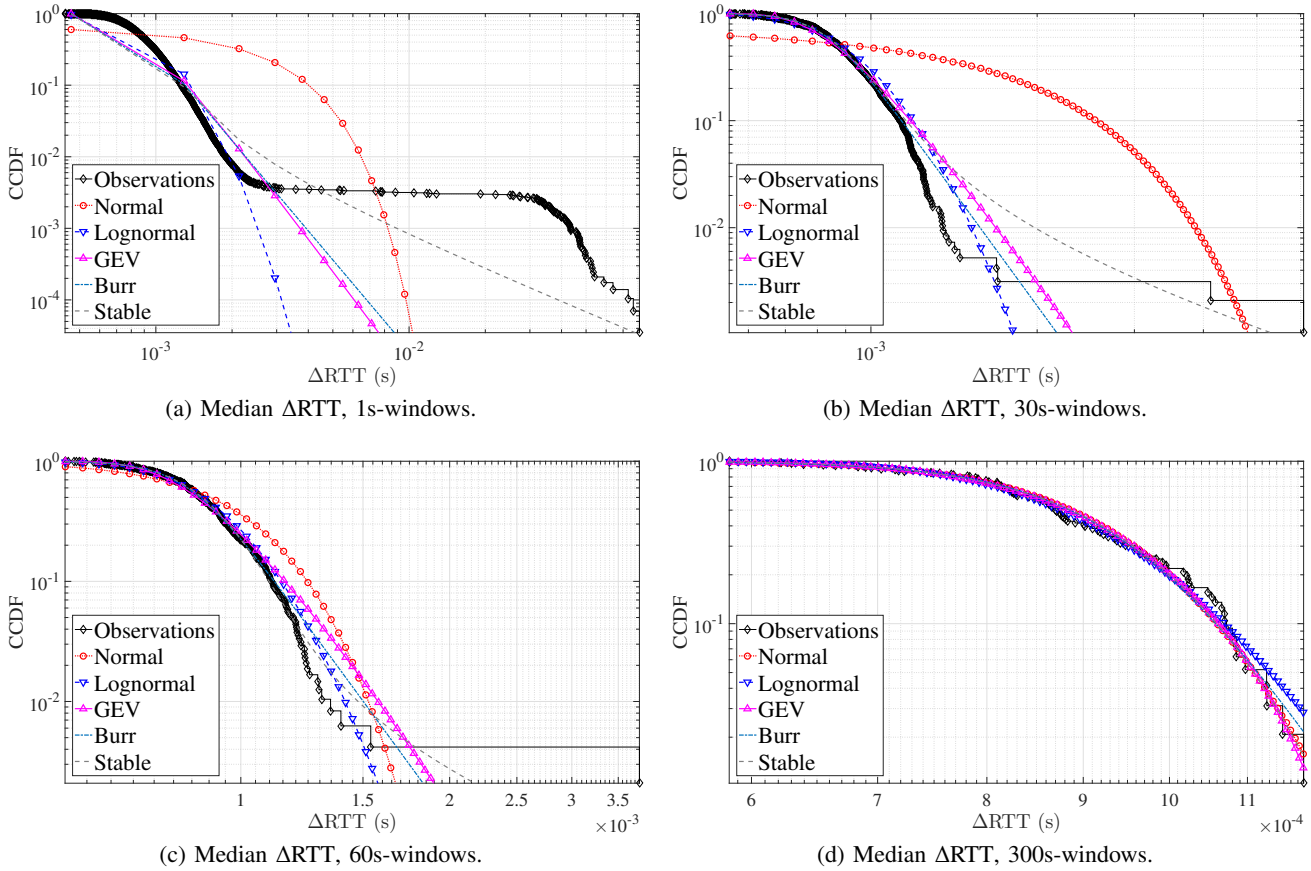
On the other hand, extreme values' modeling can be tackled using other order statistics instead of the median. This is useful to represent boundaries for network equipment performance improving the detection of service disruptions. Hereinafter, we consider the 95$^{\text{th}}$ percentile for illustrative purposes and aiming at the discrimination of the large ΔRTT peaks in our data. Time-based pre-filtering with the later statistical modeling allows the model to capture extreme values with a reduction of over- or under-represented atypical observations. The effect of window size in this procedure is illustrated in Figure 9, which shows the convergence to a stable situation with an acceptable fitting of the extreme values and a progressive reduction of the weight of observations near central values.

With this, AdPRISMA corroborated its capabilities to reach a comprehensive yet simple description of how network elements behave. Remarkably, this description distinguishes the dynamics of central and extreme values and includes quantitative criteria to balance variance—i.e., reducing the effect of bursty measurements—and bias—i.e., considering the ECDFs' variability during the projection stage.

V. Discussion

The evaluation of AdPRISMA has illustrated the viability of monitoring systems with the desirable characteristics that grounded this work. Our proof of concept and case studies have exposed some remarkable ideas that improve current network management state of the art:

1) *Passive retrieval of relevant information can be distributed:* AdPRISMA implements a distributed data gathering strategy, which is useful to improve the scalability of monitoring systems. Then, data aggregation and



(a) Median ΔRTT, 1s-windows.

(b) Median ΔRTT, 30s-windows.

(c) Median ΔRTT, 60s-windows.

(d) Median ΔRTT, 300s-windows.

Fig. 8: Results for ΔRTT in *Dataset*$_2$. Time-based aggregation, median projection, with diverse window sizes.

processing provided meaningful contextual information to characterize the network state comprehensively.

2) *RTT components help to locate where performance issues are most likely to appear:* as shown above, the observations of RTT do not fully characterize the behavior of RTT components. Therefore, the application of strategies such as ours can improve the detection and actuation in case of network issues.
3) *Models that are more complex are not necessarily better:* our evaluation and first case study reveal that simpler models may be better to represent measurements if complexity is included in the selection criteria. That is, slight improvements of goodness of fit may not justify the usage of more sophisticated models.
4) *Projection of flow-based estimates can improve the extraction of node-level KPIs:* second case study presented AdPRISMA outcomes when analyzing data lasting for several hours and in the presence of large peaks in the RTT component under test. This has exposed a moral: since single-flow estimates can produce sub-optimal models with high variance, we need techniques that reduce the variance and allow us the characterization of the regular behavior in the vantage point correctly.

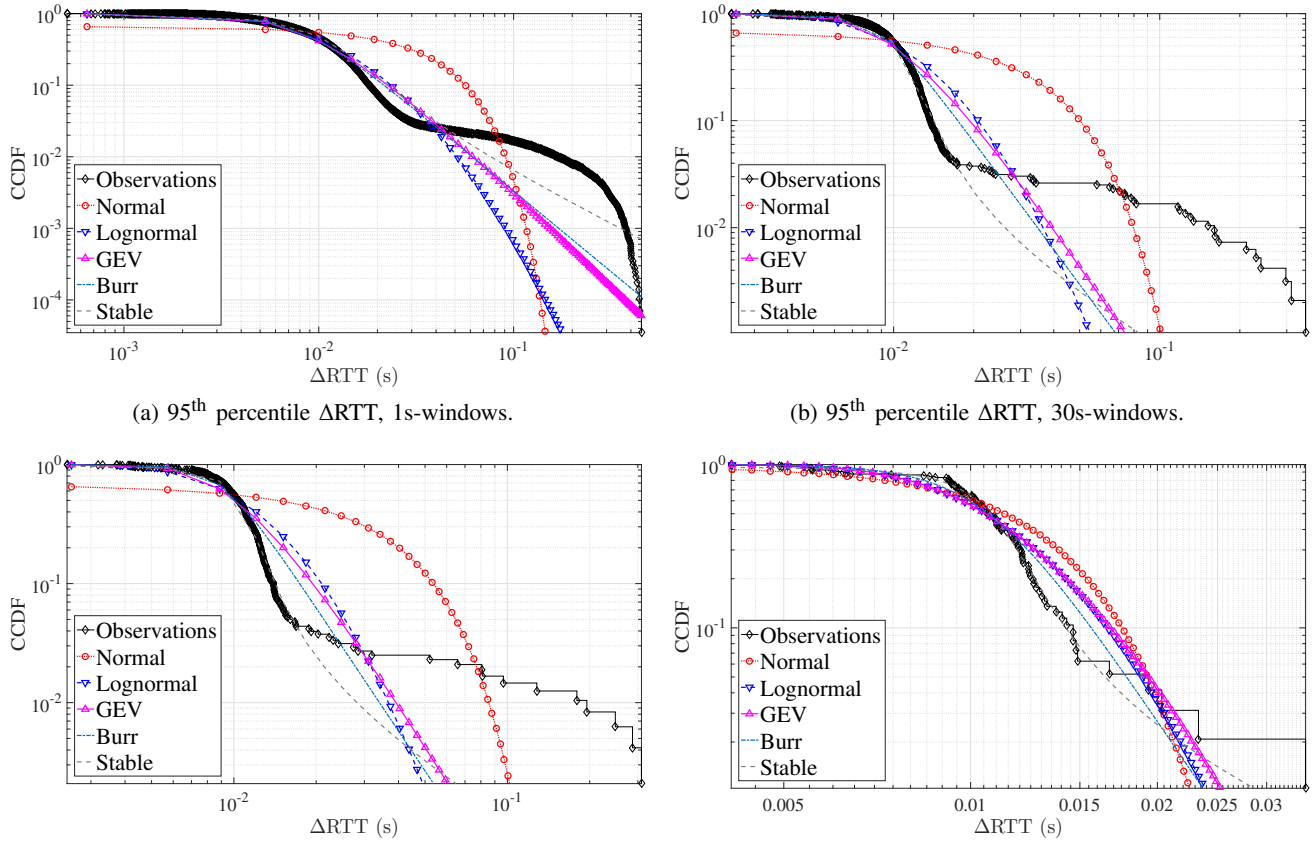
However, some practical issues may arise during the operation of AdPRISMA. For instance, random packet sampling in vantage points may harm the fitting of models because of the reduction of mutual observations.

VI. Conclusion

We have described AdPRISMA, a network monitoring system able to provide comprehensive multi-point RTT modeling. It relies on the decomposition of passive RTT values in components that reflect the state of different network segments. AdPRISMA is equipped with an automatic model selection algorithm that takes into account goodness of fit and complexity to optimize computational cost of analysis. This fitting also includes projection methods to improve the extraction of KPI trends from single-flow estimates.

Although experimental results have focused on RTT measurements, our methodology can be extended to other performance indicators measured at multiple points—e.g. delay variation or jitter at each vantage point. Specifically, AdPRISMA provides a processing engine with a general set of features for measurements: namely, (a) pre-process, correlate and cluster the measurements, (b) segment observations using time or spatial location, and (c) fit models and choose the most suitable one depending on the situation and trade-offs between accuracy and complexity. Furthermore, AdPRISMA's design and operation make easier the definition of wide monitoring perspectives, as observations from different vantage points can be simultaneously considered and correlated.

These features turn AdPRISMA into a promising framework to enrich network management platforms and tools, given its advantages for the characterization of network KPIs with high adaptability. For instance, high values can be distin-



(a) 95th percentile ΔRTT, 1s-windows.

(b) 95th percentile ΔRTT, 30s-windows.

(c) 95th percentile ΔRTT, 60s-windows.

(d) 95th percentile ΔRTT, 300s-windows.

Fig. 9: Results for ΔRTT in *Dataset*$_2$. Time-based aggregation, 95th percentile projection, with diverse window sizes.

guished from atypical values—as seen in the first case study—, and projections with different window sizes can be used—as shown in the second case study—which may be helpful to improve bias-variance trade-offs. Remarkably, both use cases were intended to illustrate how these insights can improve and support the business logic inherent to many management tasks. Hereby, we believe that our work provides evidences of AdPRISMA applicability to the monitoring, analysis and modeling of diverse network KPIs.

In sum, the experimental assessment of our proof of concept exposed that it provides promising results both in synthetic scenarios and in field trials with real-world traces gathered from enterprise networks. Additionally, we have released a prototype that is freely available to the community.[2]

REFERENCES

[1] D. Perdices, D. Muelas, L. de Pedro, and J. E. López de Vergara, "Network Performance Monitoring with Flexible Models of Multi-Point Passive Measurements," in *Proc. 14th International Conference on Network and Service Management (CNSM)*, Nov 2018, pp. 1–9.

[2] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 909–928, Second 2013.

[3] V. Uceda, M. Rodríguez, J. Ramos, J. L. García-Dorado, and J. Aracil, "Selective Capping of Packet Payloads at Multi-Gb/s Rates," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1807–1818, June 2016.

[2]https://github.com/hpcn-uam/adprisma

[4] J. Sonchack, A. J. Aviv, E. Keller, and J. M. Smith, "Turboflow: Information rich flow record generation on commodity switches," in *Proc. 13th EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 11:1–11:16.

[5] D. Muelas, J. E. López de Vergara, J. R. Berrendero, J. Ramos, and J. Aracil, "Facing network management challenges with functional data analysis: Techniques & opportunities," *Mobile Networks and Applications*, vol. 22, no. 6, pp. 1124–1136, Dec 2017.

[6] F. Ricciato, F. Vacirca, and P. Svoboda, "Diagnosis of capacity bottlenecks via passive monitoring in 3g networks: An empirical analysis," *Computer Networks*, vol. 51, no. 4, pp. 1205 – 1231, 2007.

[7] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato, and M. Rupp, "A comparison between one-way delays in operating HSPA and LTE networks," in *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, May 2012, pp. 286–292.

[8] T. Kirschstein, S. Liebscher, G. Porzio, and G. Ragozini, "Minimum volume peeling: A robust nonparametric estimator of the multivariate mode," *Computational Statistics & Data Analysis*, vol. 93, pp. 456 – 468, 2016.

[9] J. Mandel, *The Statistical Analysis of Experimental Data*. Dover Publications, 1984.

[10] J. Liebeherr, A. Burchard, and F. Ciucu, "Delay bounds in communication networks with heavy-tailed and self-similar traffic," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1010–1024, Feb 2012.

[11] F. Simmross-Wattenberg, J. I. Asensio-Pérez, P. Casaseca-de-la-Higuera, M. Martín-Fernández, I. A. Dimitriadis, and C. Alberola-López, "Anomaly detection in network traffic based on statistical inference and alpha-stable modeling," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 494–509, July 2011.

[12] E. Carisimo, S. P. Grynberg, and J. Alvarez-Hamelin, "Influence of traffic in the stochastic behavior of latency," in *TMA PhD school*, 2017.

[13] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network,"

IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, pp. 908–921, Aug 2003.
[14] J. A. Hernández and I. W. Phillips, "Weibull mixture model to characterise end-to-end Internet delay at coarse time-scales," *IEE Proceedings - Communications*, vol. 153, pp. 295–304(9), April 2006.
[15] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An introduction to statistical modeling of extreme values*. Springer, 2001, vol. 208.
[16] P. R. Tadikamalla, "A Look at the Burr and Related Distributions," *International Statistical Review / Revue Internationale de Statistique*, vol. 48, no. 3, pp. 337–344, 1980.
[17] J. Royuela-del-Val, F. Simmross-Wattenberg, and C. Alberola-López, "libstable: Fast, Parallel, and High-Precision Computation of α-Stable Distributions in R, C/C++, and MATLAB," *Journal of Statistical Software*, vol. 78, no. i01, 2017.
[18] G. Julián-Moreno, J. E. López de Vergara, I. González, L. de Pedro, J. Royuela-del-Val, and F. Simmross-Wattenberg, "Fast parallel α-stable distribution function evaluation and parameter estimation using OpenCL in GPGPUs," *Statistics and Computing*, vol. 27, no. 5, pp. 1365–1382, Sep 2017.
[19] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, "Bridging Router Performance and Queuing Theory," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 355–366, Jun. 2004.
[20] F. Mata, J. L. García-Dorado, and J. Aracil, "Detection of traffic changes in large-scale backbone networks: The case of the Spanish academic network," *Computer Networks*, vol. 56, no. 2, pp. 686 – 702, 2012.
[21] D. Muelas, M. Gordo, J. L. García-Dorado, and J. E. López de Vergara, "Dictyogram: A statistical approach for the definition and visualization of network flow categories," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 219–227.
[22] K. Xu and F. Wang, "Cooperative monitoring for internet data centers," in *2008 IEEE International Performance, Computing and Communications Conference*, Dec 2008, pp. 111–118.
[23] A. Gupta, R. Birkner, M. Canini, N. Feamster, C. Mac-Stoker, and W. Willinger, "Network monitoring as a streaming analytics problem," in *Proc. 15th ACM Workshop on Hot Topics in Networks*, ser. HotNets '16. New York, NY, USA: ACM, 2016, pp. 106–112.
[24] F. Moradi, C. Flinta, A. Johnsson, and C. Meirosu, "ConMon: An automated container based network performance monitoring system," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 54–62.
[25] K. Holger, D. Sevil, P. Manuel, G. Alex, B. Michael, R. Aurora, M. Josep, S. M. Shuaib, vanRossem Steven, T. Wouter, and X. George, "DevOps for network function virtualisation: an architectural approach," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1206–1215, 2016.
[26] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information RFC 7011," 2013.
[27] M. Trevisan, A. Finamore, M. Mellia, M. Munafo, and D. Rossi, "Traffic Analysis with Off-the-Shelf Hardware: Challenges and Lessons Learned," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 163–169, March 2017.
[28] R. Leira, G. Julián-Moreno, I. González, F. J. Gómez-Arribas, and J. E. López de Vergara, "Performance assessment of 40 Gbit/s off-the-shelf network cards for virtual network probes in 5G networks," *Computer Networks*, vol. 152, pp. 133–143, 2019.
[29] J. Kilpi and I. Norros, "Testing the gaussian approximation of aggregate traffic," in *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 49–61.
[30] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec 1974.
[31] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 03 1978.
[32] J. A. Wellner *et al.*, "A Glivenko-Cantelli theorem and strong laws of large numbers for functions of order statistics," *The Annals of Statistics*, vol. 5, no. 3, pp. 473–480, 1977.
[33] D. R. Bickel and R. Frühwirth, "On a fast, robust estimator of the mode: Comparisons to other robust estimators with applications," *Computational Statistics & Data Analysis*, vol. 50, no. 12, pp. 3500 – 3530, 2006.
[34] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proc. 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 253–264.
[35] J. Yan and D. Jin, "VT-Mininet: Virtual-time-enabled Mininet for Scalable and Accurate Software-Define Network Emulation," in *Proc. 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, ser. SOSR '15. New York, NY, USA: ACM, 2015, pp. 27:1–27:7.
[36] D. Perdices, J. E. López de Vergara, P. Roquero, C. Vega, and J. Aracil, "FlexiTop: a flexible and scalable network monitoring system for Over-The-Top services," *Network Protocols and Algorithms*, vol. 9, no. 3-4, 2017.

Daniel Perdices (daniel.perdices@naudit.es) is research and development engineer at Naudit HPCN (Spain). He received the BSc. (Hons) degrees in Mathematics and in Computer Science (2018), and currently is a M.Sc. candidate in Mathematics and Applications, and in Information and Communications Technologies, all at Universidad Autónoma de Madrid (Spain). He researches on statistics, network traffic analysis and SDN.

David Muelas (dav.muelas@uam.es) is currently data scientist at BBVA Data & Analytics. Previously, he was researcher at Universidad Autónoma de Madrid (Spain) with interest in network traffic analysis, SDN and applied mathematics. He received the degrees in Mathematics and Computer Science (2013), M.Sc. degrees in Mathematics and Applications, and in Information and Communications Technologies (2015), and Ph.D. in Computer Science and Telecommunication Engineering (2019) all of them from Universidad Autónoma de Madrid.

Iria Prieto (iria.prieto@naudit.es) is senior analyst at Naudit HPCN (Spain). She received her M.Sc in Computer Technologies in 2011 and her Ph.D. in Computer Science in 2016, both from Universidad Pública de Navarra (Spain). At Naudit, she works on network service performance and availability monitoring, applying her research to large company networks.

Luis de Pedro (luis.depedro@uam.es) is part-time professor at Universidad Autónoma de Madrid (Spain), and president of Naudit HPCN, a company devoted to high performance traffic monitoring and analysis. He received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from Universidad Politécnica de Madrid (Spain) in 1987 and 1992, respectively. He currently researches on statistical models for network traffic.

Jorge E. López de Vergara (jorge.lopez_vergara@uam.es) is associate professor at Universidad Autónoma de Madrid (Spain), and founding partner of Naudit HPCN, a company devoted to high performance traffic monitoring and analysis. He received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from Universidad Politécnica de Madrid (Spain) in 1998 and 2003, respectively. He researches on network and service management and monitoring, having co-authored more than 100 scientific papers on this topic. He is principal investigator of TRÁFICA project at UAM and also leads the tasks assigned to Naudit HPCN at METRO-HAUL project.