Network Performance Monitoring with Flexible Models of Multi-Point Passive Measurements

Daniel Perdices[1,2], David Muelas[1], Luis de Pedro[1,2], Jorge E. López de Vergara[1,2]
[1]*Dept. Electronics and Communication Technologies*
Escuela Politécnica Superior, Universidad Autónoma de Madrid
Madrid, Spain
[2]*Naudit High Performance Computing and Networking, S.L.*
Madrid, Spain
daniel.perdices@naudit.es, {dav.muelas, luis.depedro, jorge.lopez_vergara}@uam.es

***Abstract*—Many management actions for networking infrastructures require to simultaneously consider the state of several network elements. This is particularly critical in the case of reconfigurable deployments, such as Virtual or Software-Defined Networks, to scale the affected equipment up and prevent performance bottlenecks. In this light, we present dPRISMA (distributed Passive Retrieval of Information, and Statistical Multipoint Analysis), a passive monitoring system intended to fit statistical models for network measurements and raise alarms in the case of extreme behaviors. As distinguishing features, dPRISMA relies on cost-effective multi-point network measurements, and is able to select a suitable parametric model optimizing the trade-off between fitting and complexity. Therefore, it can (i) correlate records collected from several vantage points and detect where performance issues are most likely to appear; (ii) adjust alarms in terms of the probability of events; and (iii) adapt its behavior to dynamic network conditions while presenting a fair identification of anomalous situations. We evaluate dPRISMA with experiments both in virtual environments and with real-world data to provide evidences of its applicability.**

***Index Terms*—network monitoring, probability, passive measurements, performance management, pro-active management**

I. Introduction

In recent times, network environments have turned from mostly static infrastructures to a challenging context where flexible software-based configurations are becoming common. On this basis, 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 in a flexible manner —*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 [1].

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, network monitoring efforts are related to the thinning and capping of network data [2], and the exploitation of the distributed nature of these data to shift part of the analysis to the network equipment [3].

Moreover, network operation requires robust analysis of network measurements to select 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 [4]. 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 network conditions and spurious values can lead to

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.

978-3-903176-14-0

TABLE I: Parametric models included in dPRISMA.

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$(\alpha,\beta,\mu,\sigma)$	No closed formula	No closed formula

biased decisions. Models should consider a compromise between goodness of fit and complexity, to optimize analytics and prevent unnecessary computational costs.

In line with these trends, we present the design of dPRISMA, a system 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 [5], [6].

Our proposal integrates several models that proved to characterize the distribution of RTT fairly: dPRISMA automatically ranks those models, and select the one with the highest goodness of fit and lowest complexity. With this, it 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 [7].

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 dPRISMA 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 dPRISMA, 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 dPRISMA. We start with a review of statistical models for RTT measurements, to justify the selection of the models in our system. Then, we focus on other 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 [8], 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 [9]–[11]. 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 [12], 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 [13], 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 [14], 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 [15] and preliminary results in other applications to network data [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 [11]. 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. Multi-point distributed monitoring systems

Network slicing and virtual networks on top of shared hardware require flexible and scalable 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 [19] 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 [1]. The architecture of dPRISMA 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 [3] is a recent proposal that relies on the embedding of flow generation into programmable switches. However, the 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 [20] distributes monitoring tasks to different network elements, providing a query-based API that can be exploited by other modules. Parallel to these proposals, dPRISMA provides high-level analytics after aggregation and correlation of traffic packets or flow records that may be produced by different sources and methodologies.

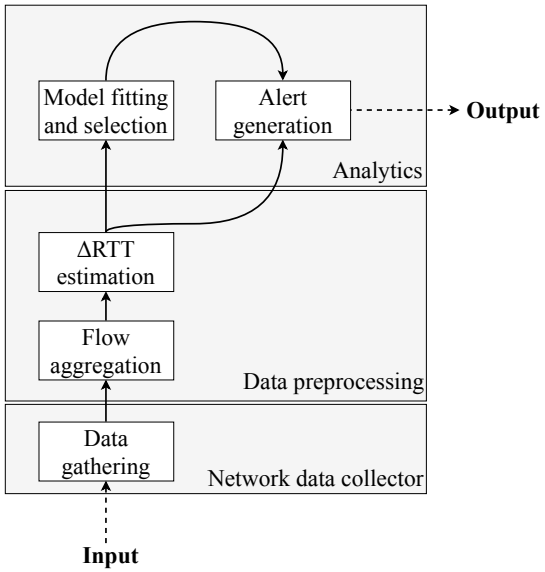


Fig. 1: Functional modules of dPRISMA.

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 [21]. The modular design of dPRISMA 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 [22].

III. System architecture and description

Along this section, we describe the main functional components of dPRISMA, which are summarized in Figure 1. In the current proof of concept implementation, dPRISMA 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 [23].

Additionally, we synthesize the operation of dPRISMA in Figure 2. First of all, passive measurements are gathered from the available vantage points. These measurements are aggregated in dPRISMA, 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

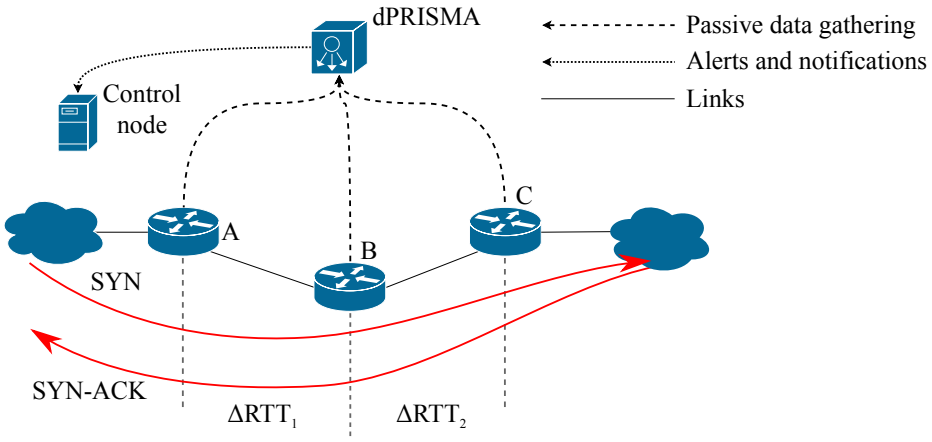


Fig. 2: Operation of dPRISMA. Red arrows represent a sample connection traversing the three monitored points of the network, distinguishing the different RTT components that are estimated to detect possible bottlenecks.

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 [23], 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 40Gb/s in commodity hardware —e.g., see [24], [25]. 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 this operations.

dPRISMA 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 Figure 2, we observe that RTT in hop j is given by Equation 1:

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

By inverting this linear operator, we can obtain an estimation of the component in the network segment between vantage point j and $j+1$ with the expression in Equation 2:

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

Algorithm 1 Flow aggregation.

```
1:  function getSuperflows(flows...)
2:  table ← InitializeSuperFlowsTable()
3:  for flow in flows do
4:      if flow is ip and tcp then
5:          if flow.srcPort < flow.dstPort or
            (flow.srcPort = flow.dstPort and
            (flow.srcIp < flow.dstIp)) then
6:              quintuple ← (flow.srcIp, flow.srcPort, flow.dstIp,
                flow.dstPort, flow.ipProto)
7:          else
8:              quintuple ← (flow.dstIP, flow.dstPort, flow.srcIp,
                flow.srcPort, flow.ipProto)
9:          end if
10:         table[quintuple].addFlow(flow)
11:     end if
12: end for
13: return(table)
```

Note that these estimations do not require synchronization of equipment clocks.

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 dPRISMA 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

TABLE II: Summary of metrics for model selection.

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

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 [26], 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.

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 $AIC = 2(k - \log(\hat{L}))$ where k is the number of parameters of the model and $\hat{L}$ is the maximum of the likelihood function [27]. It is remarkable that complexity is just evaluated with the number of parameters, and this makes it a really 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 avoid overfitting in parametric models, so that the complexity and goodness of fit are balanced [28]. The expression is $BIC = \log(N)k - 2\log(\hat{L})$, 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.

C. Mode estimation

The mode of a sample is a prominent centrality measure that returns the most probable value of a distribution. 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 [7]— 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, $\widehat{\text{Mode}}(X) = argmax_{x \in \mathbb{R}} \hat{f}(x)$. 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 [29]. 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 dPRISMA proof of concept encompassed two different stages: a first one, with laboratory experiments, where we tested the system in a controlled environment; and a second one with real data coming from a data center.

The first stage was accomplished in an emulation-based experimental environment[1] on top of mininet [30], [31]. 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.

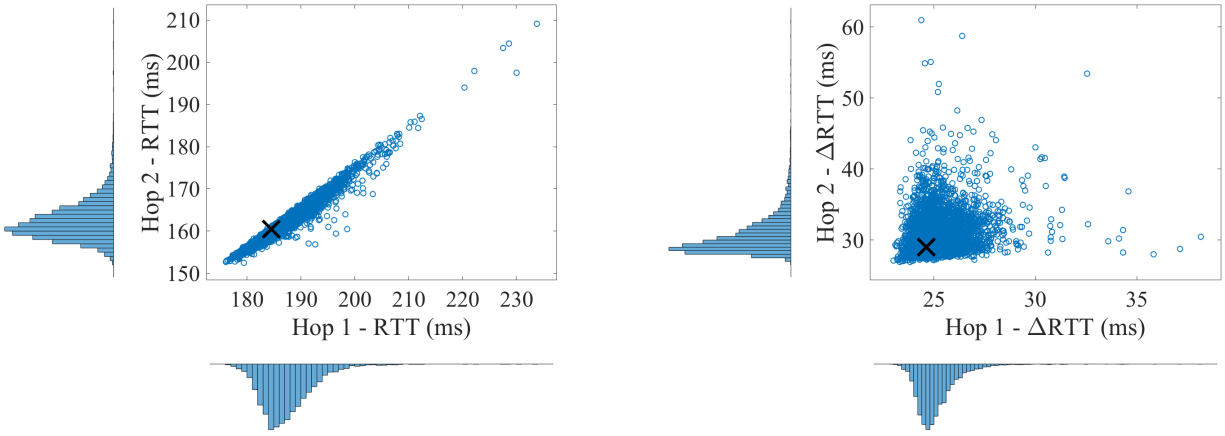
In order to make this situation closer to a real network, background traffic is introduced. Several techniques were used to generate such load: (i) ICMP ping with random intervals, (ii) `iperf` and (iii) traffic generators that rely on TCP connections to a conventional TCP or HTTP server [32].

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 dPRISMA to assess its outcomes in an actual case study. This dataset, hereinafter denoted *Dataset$_{ISP}$*, 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.

[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. 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**	$\mathbf{R^2}$	**AIC**	**BIC**	**Mode**	$\mathbf{R^2}$	**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

- 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.

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.

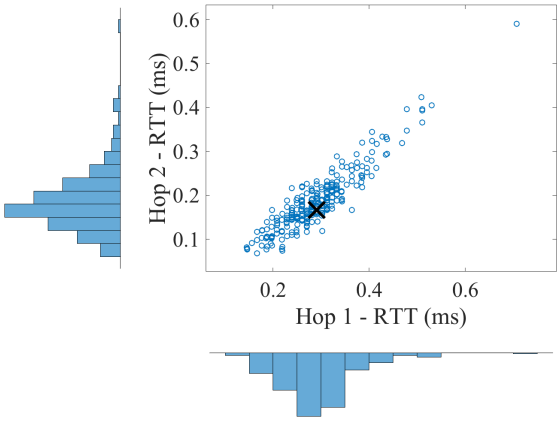
Figure 3 shows scatter plots of the RTT and ΔRTT in the two first hops —there were no significant differences with measurements in the other vantage points, so we omit the consideration of every combination for the sake of brevity. After truncating extreme values to improve visualization, the scatter 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 dPRISMA.

1) Model fitting and selection: 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. Though, **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 dPRISMA can be tuned to take into account and balance several factors (complexity, number of samples or just goodness-of-fit) depending on the context.

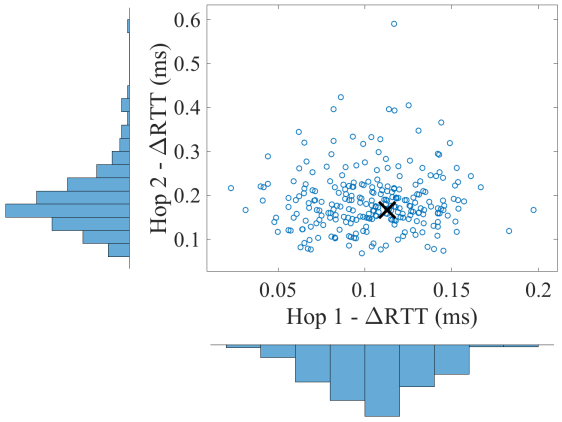
2) Mode estimation: Table III also includes estimated modes, including the computations with KDE and HSM. In each case, we observe than the mode is around the theoretical expected values, both for ΔRTT_1 and ΔRTT_2 (24 and 28ms, respectively) plus an additional delay of ∼0.7 ms because of the background 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. Case study: analysis of a data center network

Once we have assessed the accuracy of dPRISMA, we inspect the results obtained during the study of a real data center network. Similarly to the previous experiments, we present scatter plots of RTT and ΔRTT in Figure 4, and summarize the results of model fitting and mode estimation



(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$_{ISP}$*. The $\times$ shows the intersection of the modes of hops 1 and 2.

TABLE IV: Estimated mode of ΔRTT_1 and ΔRTT_2 in *Dataset$_{ISP}$*, 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	$\sim$0.1042ms	0.991	-4750.721	-4282.579	$\sim$0.1568ms	**0.970**	**-4296.883**	**-4282.579**

in Table IV. Additionally, we include in Figure 5 the representation of sample data compared to the three models that provided the best goodness of fit. Figures 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 5th and 95th 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 Figure 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.

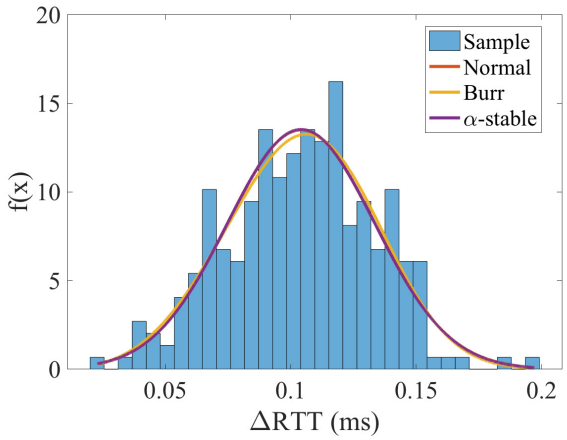
However, the behavior of ΔRTT_2 (*i.e.*, measurements in the second vantage point) is very different. In this case, the preferred model is the α-stable distribution, with better scores (either when considering $\mathbf{R^2}$, **AIC** or **BIC**) for any other option. The skewness and tail of ΔRTT_2 prevent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

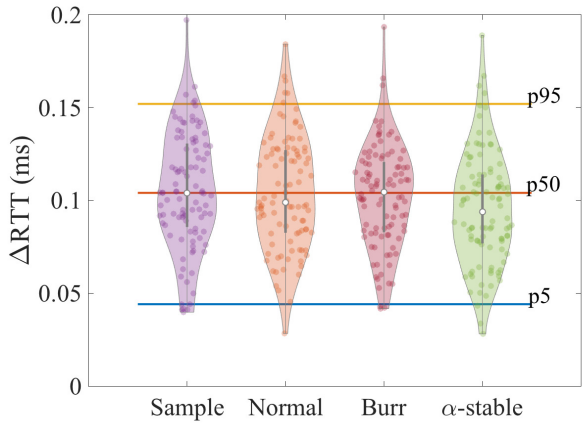
V. Discussion

The evaluation of dPRISMA has illustrated the viability of monitoring systems with the desirable characteristics that grounded this work. Our proof of concept and case study has exposed some remarkable ideas:

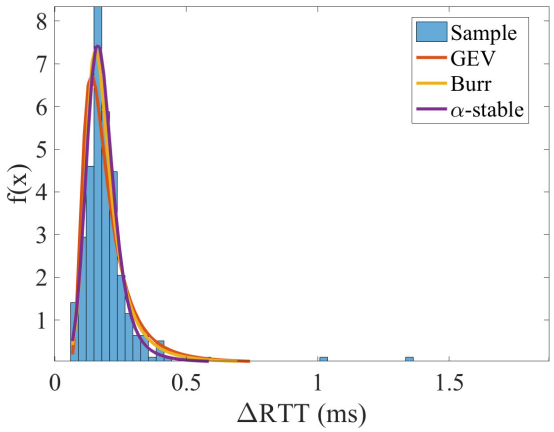
1) *Passive retrieval of relevant information can be distributed:* dPRISMA implements a distributed data gathering strategy, which is useful to improve the scalability of monitoring systems. Then, data aggregation and processing provided meaningful contextual information to characterize the network state 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.



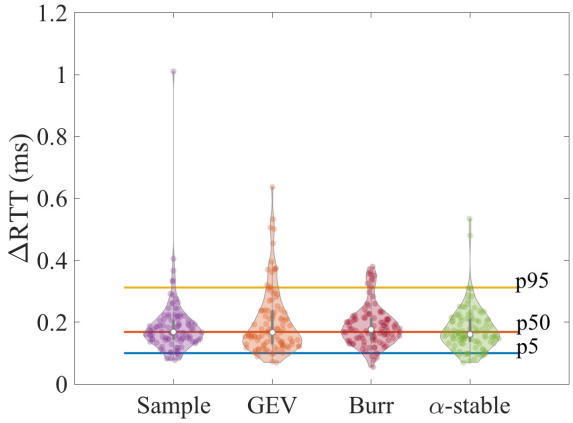
(a) Histogram and density comparison, ΔRTT_1.



(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 $Dataset_{ISP}$.

3) *More complex models are not necessarily better:* our evaluation and 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 application of more sophisticated models.

However, some practical issues may arise during the operation of dPRISMA. For instance, random packet sampling in observation points may harm the fitting of models. That is, packet sampling can reduce the applicability of our strategy for RTT estimation and decomposition. This applies when gathering NetFlow or IPFIX records, which can potentially limit the deployment of dPRISMA in these scenarios. However, we plan to further analyze to what extent this can jeopardize the results of this type of strategies, and how to overcome such limitations with novel capabilities of network equipment.

VI. Conclusion

Along this work, we have described a system for network monitoring able to provide comprehensive multi-point analysis of RTT values. It relies on the decomposition of RTT values in different components that reflect the state of different network segments. We have equipped such a system with an automatic model selection algorithm that takes into account goodness of fit and complexity, to optimize computational cost of the analysis of passive measurements. The experimental assessment of our proof of concept exposed that it provides promising results both in synthetic scenarios and in field trials with data from a data center network. Additionally, we have released a prototype that is freely available to the community.[2]

Nonetheless, some future work lines are still open before unveiling the capabilities of such an approach. First, we plan to extend data gathering modules to improve interoperability with SDN and virtualized elements. Additionally, and as stated above, we are starting to study the compatibility of dPRISMA with packet sampling techniques to alleviate computational burdens. Finally, we point to the exploration of RTT decomposition as predictor of network overloads and failures.

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

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An introduction to statistical modeling of extreme values*. Springer, 2001, vol. 208.

[15] 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.

[16] C. Mancha Díaz and L. de Pedro Sánchez (supervisor), "Análisis de modelos estadísticos de tráfico en Internet," Bachelor Thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 2018.

[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] 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.

[20] A. Gupta, R. Birkner, M. Canini, N. Feamster, C. Mac-Stoker, and W. Willinger, "Network monitoring as a streaming analytics problem," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, ser. HotNets '16. New York, NY, USA: ACM, 2016, pp. 106–112.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] G. Julián-Moreno, R. Leira, J. E. López de Vergara, F. J. Gómez-Arribas, and I. González, "On the feasibility of 40 gbps network data capture and retention with general purpose hardware," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ser. SAC '18. New York, NY, USA: ACM, 2018, pp. 970–978. [Online]. Available: http://doi.acm.org/10.1145/3167132.3167238

[26] J. Kilpi and I. Norros, "Testing the gaussian approximation of aggregate traffic," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 49–61.

[27] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec 1974.

[28] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 03 1978.

[29] 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.

[30] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 253–264.

[31] J. Yan and D. Jin, "VT-Mininet: Virtual-time-enabled Mininet for Scalable and Accurate Software-Define Network Emulation," in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, ser. SOSR '15. New York, NY, USA: ACM, 2015, pp. 27:1–27:7.

[32] 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.