Cooperative Learning for Disaggregated Delay Modeling in Multidomain Networks

F. Tabatabaeimehr, M. Ruiz, C.-Y. Liu, X. Chen, R. Proietti, S. J. B. Yoo, and L. Velasco

***Abstract*—Accurate delay estimation is one of the enablers of future network connectivity services, as it facilitates the application layer to anticipate network performance. If such connectivity services require isolation (slicing), such delay estimation should not be limited to a maximum value defined in the Service Level Agreement, but to a finer-grained description of the expected delay in the form of, e.g., a continuous function of the load. Obtaining accurate end-to-end (e2e) delay modeling is even more challenging in a multi-operator (Multi-AS) scenario, where the provisioning of e2e connectivity services is provided across heterogeneous multi-operator (Multi-AS or just domains) networks. In this work, we propose a collaborative environment, where each domain Software Defined Networking (SDN) controller models intra-domain delay components of inter-domain paths and share those models with a broker system providing the e2e connectivity services. The broker, in turn, models the delay of inter-domain links based on e2e monitoring and the received intra-domain models. Exhaustive simulation results show that composing e2e models as the summation of intra-domain network and inter-domain link delay models provides many benefits and increasing performance over the models obtained from e2e measurements.**

***Index Terms*— Cooperative Learning; Multidomain Networks; End-to-end Delay.**

I. Introduction

TOGETHER with throughput, one of the key performance indicators of packet connectivity services is end-to-end (e2e) delay. In fact, e2e delay plays a particularly important role in the development of new networking solutions and is one of the main drivers for the development of 5G and beyond networking [1]. Therefore, the maximum delay is one of the parameters to be guaranteed and it is part of the Quality of Service (QoS) requirements that customers request at the provisioning phase of packet connections.

Manuscript received November 30, 2020.

The research leading to these results has received funding from the AEI/FEDER TWINS project (TEC2017-90097-R), from the Catalan Institution for Research and Advanced Studies (ICREA), and from US-EU research collaboration initiative funded by the NSF award #ICE-T:RC 1836921.

Fatemehsadat Tabatabaeimehr, Marc Ruiz, and Luis Velasco (luis.velasco@upc.edu) are with the Optical Communications Group at the Universitat Politècnica de Catalunya, Barcelona, Spain. Che-Yu Liu, Xiaoliang Chen, Roberto Proietti, S. J. Ben Yoo are with University of California (UC), Davis, USA.

QoS performance should be thus monitored periodically and *passive* and *active* monitoring techniques have been defined (see [2]). Passive monitoring entails collecting counters from routers and passive *probes*. Such data can be used for many purposed, e.g., for detecting bottlenecks, so that decisions can be made at the network management, as in [3]. In contrast, in *active monitoring*, *active probes* measure the round-trip time of packets by injecting trains of numbered and timestamped packets that are looped back by the remote probe [4]. In-band network telemetry (INT) [5] is an alternative active monitoring technique, where intermediate routers add statistics to packets so e2e QoS measurement can be achieved.

Independently of the technique, monitoring data can be stored in a centralized repository, in the Monitoring and Data Analytics system of the network operator [6] besides the Software Defined Networking (SDN) controller. On their side, customers can get monitoring data from the operator, as in [7], only in case in case of single-operator scenarios) or they can install their own active probes and used to measure e2e delay. Monitoring data can be used for the estimation of the performance of packet connections, e.g., by training a Machine Learning (ML) model [8]. Note that such estimation would facilitate SDN and customer applications operation.

In multi-operator (multi-Autonomous System –AS) networks, the provisioning of customer e2e connectivity services is provided across heterogeneous operator networks (or just *domains*). In this scenario, obtaining accurate e2e delay modeling is a challenging task that is difficult to achieve with current architectures based on intra-domain and inter-domain routing protocols, as they do not provide the needed capabilities for e2e delay management [9].

In this regard, ML models can be helpful, not only for e2e delay prediction, but also to detect deviations between predictions and real measurements, which is typical in non-stationary scenarios; note that initially small deviations can derive into anomalies [10]. To correct model inaccuracies, one needs to find the source of them. In multi-operator networks however, the source of observed inaccuracies can be any of the domains and

inter-domain links that support a given connection. Therefore, a way to find the source of deviations is by building differentiated models for domains and inter-domain links, as opposite to e2e models. Nevertheless, inter-domain delay measurements are not generally available, specially to third domains or customers.

The rest of the paper is organized as follows. Section II reviews the state-of-the-art related to delay modeling and presents the main contributions of this work. The multidomain scenario is presented in Section III. Options for modeling delay are presented and include e2e and per domain modeling that can be used afterward for prediction. Section IV focuses on inter-domain link modeling, intra-domain model correction, and inaccuracy detection and localization. The discussion is supported by the results presented in Section V. Finally, Section VI concludes the paper.

II. Related Work and Contributions

In the literature, some previous works have proposed different methods for monitoring the QoS. Because using active probes requires dealing with the introduced overhead, the authors in [11] targeted at reducing the monitoring load without compromising delay accuracy, whereas the authors in [12] proposed to use active monitoring during commissioning testing to avoid such overhead. In the later work, the authors injected packet trains reproducing the real traffic that the service will support, and measurement data were used afterwards to produce a specific delay model for the packet connection that could be used during connection operation time. They showed that delay measurements can largely differ depending on the characteristics of the packet trains used for monitoring and therefore, those trains should be designed according to the traffic expected for the individual service.

Aiming at collecting fine grain measurements, INT can be adopted to collect network status hop-by-hop, which fits well in single operator SDN environments. However, precisely the hop-by-hop characteristics of INT difficulties its application on multiple operator scenarios, as it might reveal internal details of the domains. Another option is network tomography [13], which consists in inferring domain and inter-domain link delay components from e2e measurements together with the available topology and routing information. Nonetheless, precisely topology and routing information, is not generally available in multi-operator scenarios.

Regarding the application of ML on the collected monitoring data, some previous works have proposed models for short-term and long-term traffic prediction at different time scales (see, e.g., [14] and [15] for second and day time scale, respectively). ML can be also used to predict the delay, which should be bounded by the maximum delay defined at the provisioning time. For instance, neural networks and random forests are proposed in [16] to model e2e delay, where the authors used traffic matrices (generated with the NS-3 simulator assuming Gaussian distributed traffic intensity for each flow) as input of the predictive models and evaluated their robustness and accuracy under different network scenarios. Other works have proposed alternative networking models together with control and orchestration plane architectures to provide the committed delay to customer connections. The authors in [17] leveraged a single domain SDN paradigm and proposed a model able to relate the complex network relationships to produce accurate estimates of the per-packet delay distribution and loss. The authors in [18] proposed a network slicing orchestration solution able to handle e2e latency in multidomain single-operator networks. They leveraged a multi-armed-bandit method to allocate resources to slices to meet end-to-end latency requirements. Current networks are non-stationary in general and therefore, pre-trained networking models require fine-tuning to correct the model-mismatch problem, as highlighted in [19].

An issue related to the use of ML techniques is that a large data set for training is required to obtain accurate ML models. To solve that issue, authors in [12] used active monitoring to inject synthetic traffic reproducing the expected traffic patterns of a given customer connection and use a simulation tool to generate enough data for training delay models.

Multi-operator networks bring additional challenges, in particular regarding e2e delay. One possible architecture to provide e2e services is that of peer-to-peer, where operators exchange information among them directly. As an example, the authors in [20] studied the convenience of exchanging information related to the guaranteed latency and resource availability in each of the domains to reduce service provisioning blocking probability.

Instead of peer-to-peer multidomain architectures, an e2e service provider can deploy a broker system (e.g., based on the one proposed in [21]) that coordinates e2e path provisioning and relies on domain SDN controllers for intra-domain provisioning. Here, domains would share information with the broker in targeting at providing better services while receiving performance feedback from the broker. This information or knowledge sharing has been successfully applied for autonomic domain networking [22]. Nonetheless, for that sharing to be realistic, exchanged information needs to be conveniently abstracted, so to ensure that internal details of the domain are not revealed. In that regard, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

authors in [23] proposed a knowledge-based multidomain service provisioning framework, where intra-domain topologies are *abstracted* and shared with the broker for multidomain network automation tasks. A similar approach was proposed by the authors in [24], where abstracted topology was used for inter-domain connection provisioning and QoS assurance.

Taking advantage of abstracted information is not always an easy task and it might lead to poor adaptability and resource efficiency. In this context, some previous works have proposed to leverage ML to develop cognitive multidomain provisioning schemes. The authors in [25], formulated the operations of multidomain networks as a multi-agent learning system and presented a multi-agent deep reinforcement learning approach to enable domain controllers and brokers to learn cooperative provisioning policies from performed operation experiences. The authors in [26] proposed an ML model to produce inter-domain routing solutions autonomously by taking advantage of historical provisioning traces. Finally, the authors in [27] demonstrated collaborative learning schemes for accurate quality of transmission estimation in multidomain optical networks, where the distributed ML blocks deployed in the broker and domain controllers learn inter-domain QoT estimators by exchanging just necessary learning data.

In this paper, we present a novel approach for QoS modeling in multidomain networks. The approach is built on top of the principle of cooperative learning, where different entities establish a positive interdependence for the sake of a powerful collective intelligence. We propose a collaborative approach between the domains and the broker, where the domains exchange abstracted data to achieve robust and accurate ML models for e2e QoS (delay) estimation. Those models can then be used for multiple purposes, such as connectivity provisioning based on QoS estimation or connectivity reconfiguration triggered by anticipated detection of QoS degradation. Assuming that each network domain can model its internal, limited view of delay based on its own traffic and delay measurements, the broker entity plays the role of combining the different domain delay models for an e2e view. By providing models instead of data, domains can share knowledge with the broker and, at the same time, enforce privacy policies. With this approach, the broker can detect and infer inaccuracies in the partial models to trigger their retraining. Specifically, the contributions of this paper are:

- The collaborative approach between the domains and the broker, where the latter provides multidomain connectivity with QoS constraints to end customers. Domains share delay models with the broker, which composes e2e delay models as combination of domain and inter-domain link models. Such approach enables the capacity to infer delay performance before establishing an e2e path, as well as to find the cause of deviations between the model and the measurements.
- The specific procedures for: *i*) inter-domain link delay estimation based on e2e delay measurements and per domain delay models; *ii*) intra-domain model inaccuracy detection with measures to correct intra domain models; *iii*) detection of the source of the inaccuracy once it is detected.

III. End-To-End And Per-Domain Delay Estimation

Fig. 1 shows the control architecture considered in this paper. We assume a dynamic scenario where a

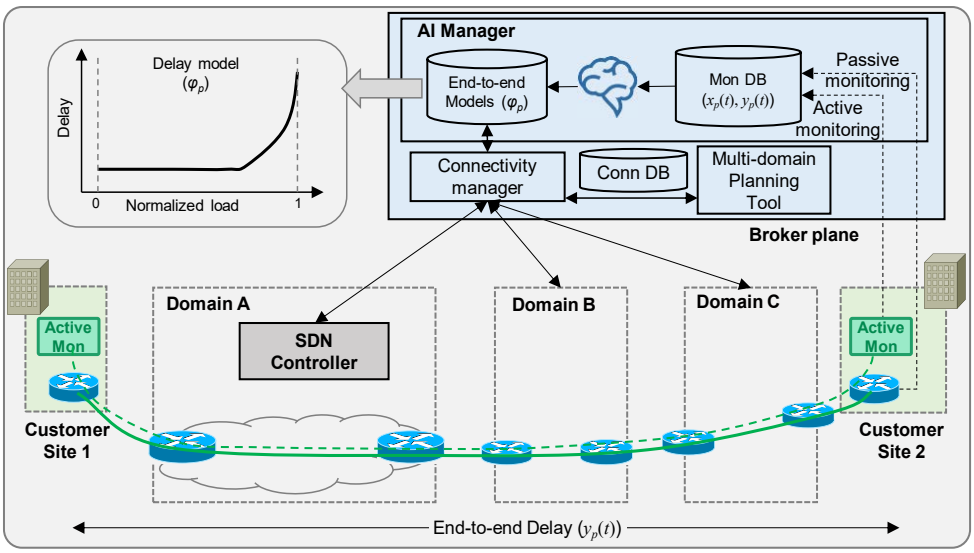

Fig. 1. Example of e2e delay and control architecture.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

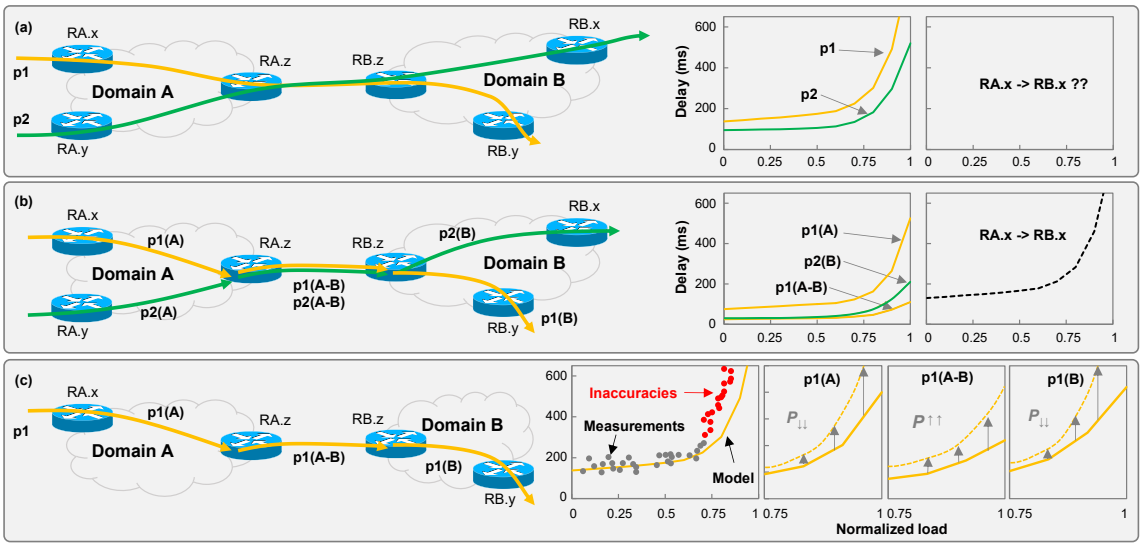

Fig. 2. Provisioning of multidomain requests: reference approach (a) vs compound approach (b). Example of inaccuracy (c).

connectivity manager in the broker receives and processes requests for connectivity with QoS requirements in terms of throughput and maximum delay among customer endpoints. The connectivity manager uses a *planning tool* for provisioning and reconfiguration purposes [28], and it is assisted by ML-based models to enhance connectivity provisioning. The broker connects to a set D of domains interconnected by a set L of inter-domain links and altogether provides connectivity to a set P of e2e connections, the performance of which is continuously monitored.

Let us assume first that the performance of each connection p is monitored e2e between the endpoints in the customer sites and that data is gathered by the broker for various purposes, like QoS analysis, modeling, etc. In particular, we assume that throughput, $x_p(t)$, and delay, $y_p(t)$, are measured periodically, e.g., every 1 min, by the customer edge routers (passive monitoring) and that active monitoring is carried out. Then, after a sufficiently large period, enough data can be collected to train ML models for every connection. Note that protocols like IPFIX, gRPC, etc. can be used for monitoring data collection.

Among possible ML models, path delay models (denoted as φ^*_p) can be used to predict a delay-related performance metric, e.g., average or maximum delay, as a function of the normalized load (computed as the ratio between the measured throughput and the capacity of the path) [12]. The embedded graph in Fig. 1 illustrates an example of delay model. End-to-end delay ML models can be used, e.g., to anticipate QoS degradation and trigger reconfiguration.

Note that the φ^*_p models not only allow analyzing the e2e delay of a single path p but they can also be used to get some insight on the performance of the domains by considering groups of paths that cross a given domain. An example is in the case of detecting model inaccuracies (e.g., significantly higher delay than expected for the observed traffic load) in a group of paths; correlation of their routes through the domains can lead to finding a common set of *segments*, either domains or inter-domain links, that could potentially hold the source of the inaccuracy. Once some segment(s) have been identified, re-routing of those affected paths could be performed to avoid them.

Nevertheless, this AI-assisted architecture suffers from an inherent drawback: the multidomain network is analyzed as a black-box, a fact that limits the applicability of advanced multidomain smart operation. An example is illustrated in Fig. 2, where two paths between domains *A* and *B* (*p1* from *RA.x* to *RB.y* and *p2* from *RA.y* to *RB.x*) are established and e2e delay models have been accurately trained after some data collection phase (Fig. 2a). Then, let us now consider that a new request for a path from *RA.x* to *RB.x* arrives. Since no path between *RA.x* and *RB.x* was previously established, the broker does not have available e2e monitoring data and delay models to predict the delay behavior of such new connectivity request. This fact limits smart provisioning decision making, e.g., to choose the best route in terms of QoS.

To overcome the aforementioned issue, we propose breaking the black-box, monolithic view of the multidomain network. End-to-end delay can be modeled by combining intra-domain and inter-domain *segment* models for those segments in the route of a path. This brings some benefits, as segment models can be used to create *compound* models not only for those established paths but also to infer models for not yet established paths. Following the previous example, the inferred model for the path request between *RA.x* to *RB.x* could be obtained by composing an e2e delay model from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

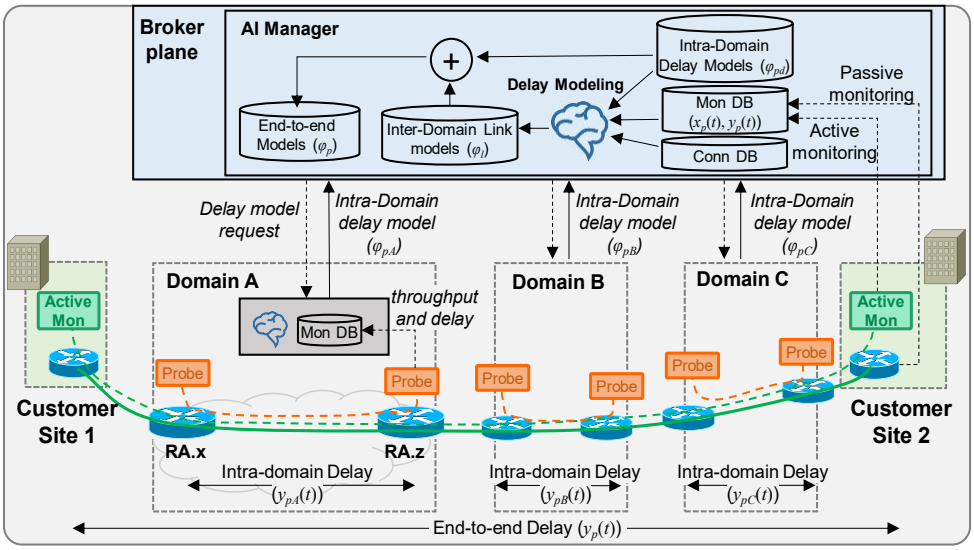

Fig. 3. Example of per-domain e2e delay and extended control architecture.

segments models: *p1*(*A*), *p1*(*A*-*B*), and *p2*(*B*) (Fig. 2b). Note that intra-domain segment models need to be computed by the domains themselves based on, e.g., active measurements carried out between the access/inter-domain interfaces for a given path in the domain. Besides, the delay introduced in inter-domain links needs also be measured, which is more difficult as active measurements should be carried out across domains. For this very reason, we target at modeling the delay of inter-domain links at the broker level thus relaxing the need of multidomain active monitoring.

Let us see now how we can identify the segment(s) that are producing delays higher than expected. Once inaccuracies between model predictions and real measurements are detected for these segments, they can be excluded for route computation in case re-routing needs to be performed. Note that the black-box approach presented above identifies common segments of affected paths by correlating routes. However, such a procedure would be imprecise and even meaningless if inaccuracies are detected in only few paths. Fig. 2c illustrates an extreme case, where inaccuracies are observed in only one path for high loads. By analyzing the models of every segment and computing how likely it is that each segment introduced such inaccuracy the most reasonable segment can be selected. With this narrower identification, more precise and proper reconfiguration actions can be taken.

The extended architecture is presented in Fig. 3, where only the details of the AI-related components are represented for the sake of clarity. *Compound* e2e delay models are composed as a combination of two distinct types of models: *i*) *intra-domain models* (φ_{pd}) and *ii*) *inter-domain link models* (φ_l). The intra-domain models are obtained by each domain controller from delay and throughput measurements collected for each of the e2e path segments carried by that domain. As an example, Fig. 3 represents an e2e path p that crosses domain A from node *RA.x* to *RA.z*; therefore, the manager in domain *A* finds the model that predicts the delay that the e2e path experiences when crossing domain *A* (φ_{pA}). To enforce domain privacy, domain models make predictions without the required knowledge from the actual domain state (e.g., the actual routes of the paths). The inter-domain models are trained by the broker, with the input of the e2e measurements and the predictions of the domain models, so that the inter-domain link components (φ_l) can be inferred. Note that, under this hierarchical training architecture, every entity (domains and broker) is free to use their modeling techniques bringing the best tradeoff between complexity and accuracy in their domains. Therefore, the combination of models into an e2e delay model needs to be done assuring that different types of models can co-exist, assuming that all delay components are functions of traffic load.

IV. Compound E2E Delay Modeling

This section details the processes that together provide accurate compound e2e delay models. Table I introduces the notation and defines the main parameters and variables that will be consistently used hereafter. Besides, as introduced in Section III, we define the compound e2e delay model of a given path p as the sum of their intra-domain and inter-domain link components, which can be formally expressed as follows:

$$\varphi_p\left(x_p(t)\right) = \sum_{d \in D} \varphi_{pd}\left(x_p(t)\right) + \sum_{l \in L} \delta_{pl} \cdot \varphi_l\left(x_l(t)\right) \quad (1)$$

Fig. 4 presents the main building blocks and their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

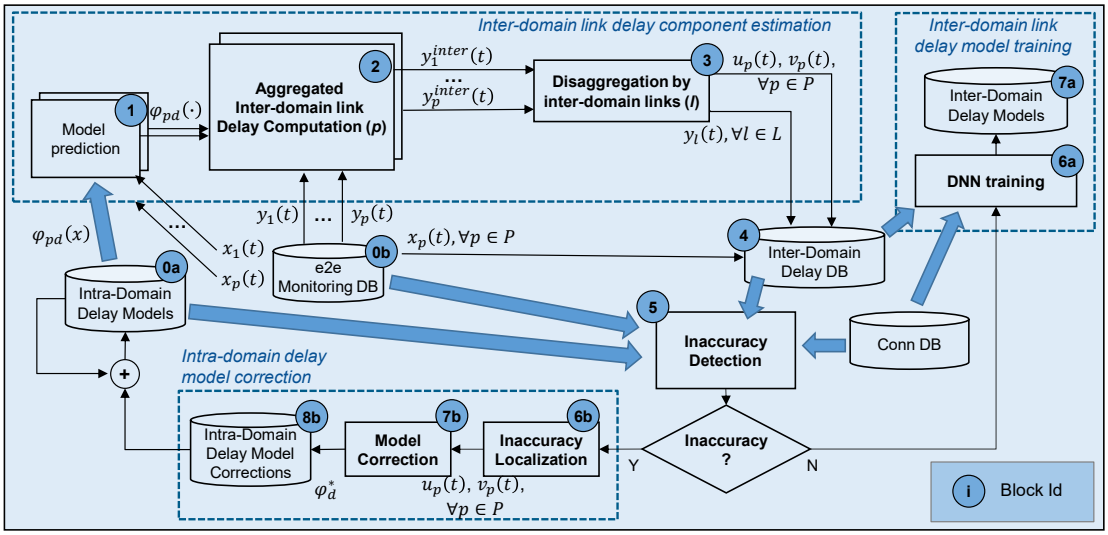

Fig. 4. Main building blocks for training and correcting delay models at the broker plane.

TABLE I. NOTATION

Sets and parameters	
P	Set of multidomain paths, index p.
$X=\{x_p\}$	Set of e2e traffic monitoring samples for all paths.
$Y=\{y_p\}$	Set of e2e delay monitoring samples for all paths.
D	Set of domains, index d.
L	Set of inter-domain links, index l.
$P(d)$	Subset of paths that traverse domain d
δ_{pl}	1 if path p traverses inter-domain link l and 0 otherwise.
$x_i(t)$	Measured traffic throughput through element i at time t, where i might identify a path p or a link l.
φ_p	Compound e2e delay model for path p
$\varphi_{pd}(x)$	Intra-domain delay model for path p in domain d. The *zero function* if p does not cross d.
φ^*_d	Correction model for intra-domain delay for domain d.
φ_l	Inter-domain link delay model for link l
Ttr	Training phase duration (in monitoring intervals)
$y_p^{inter}(t)$	Aggregated inter-domain link delay of path p at time t
$E[\cdot]$	Expectation
Decision Variables	
$y_l(t)$	Delay at inter-domain link l at time t
$\beta_{(\cdot)}(t)$	Delay bias of path/domain
$u_p(t)$, $v_p(t)$	Delay slack and surplus variables for path p

relationships for compound e2e delay modelling at the broker plane. Blocks have been conveniently numbered to facilitate the ongoing description and workflows. As depicted in Fig. 4, the main blocks can be organized into three clearly differentiated groups: *i*) inter-domain link delay component estimation (blocks 1-3); *ii*) inter-domain link delay model training (blocks 6a and 7a);and *iii*) intra-domain delay model correction (blocks 7b and 8b), which also includes inaccuracy detection (block 5) and localization procedure (block 6b) that keeps the highest goodness-of-fit through precise model improvement actions. In turn, some of the blocks perform some computation or solve optimization problems. For the sake of clarity, Algorithm I presents the pseudocode for inter-domain link delay modeling and intra-domain model correction, and Table II relates the defined blocks to optimization problems or equations. The details of the above groups and blocks are presented in the next subsections.

ALGORITHM I. INTER-DOMAIN LINK DELAY MODELING WITH INTRA-DOMAIN MODEL CORRECTION

Input: X, Y, $\{\varphi_{pd}\}$ **Output:** $\{\varphi_l\}$, $\{\varphi^*_d\}$

Corrections ← {}
while true
  $\{y_p^{inter}\}$← Inter-domain_Delay_Estimation(X, Y, $\{\varphi_{pd}\}$, *Corrections*) (*block 2*)
  $\{y_l\}, \{u_p\}, \{v_p\}$ ← Link_Delay_Disaggregation_Bias ($\{y_p^{inter}\}$, P, L) (*block 3*)
  inac ← Inaccuracy_Detection($\{u_p\}$, $\{v_p\}$) (*block 5*)
  if !*inac* **then**
    $\{\varphi_l\}$←DNN_training (X, $\{y_l\}$, P) (*block 6a*)
    break
  $d^*, \{u_p\}, \{v_p\}$←Inaccuracy_Localization($\{y_p^{inter}\}$, P,L) (*block 6b*)
  φ^*_d ← Model_Correction (d^*,$\{u_p\}$, $\{v_p\}$) (*block 7b*)
  Corrections ←*Corrections* ∪ $\{\varphi^*_d\}$
return $\{\varphi_l\}$, *Corrections*

TABLE II. RELATION BETWEEN BLOCKS AND PROBLEMS/EQS

Block	Problem	Eq(s).
2	*Inter-domain Delay Estimation*	(2)
3	*Link Delay Disaggregation*	(4)-(5)
3	*Link Delay Disaggregation Bias*	(7)-(10)
5	*Inaccuracy Detection*	(11)-(12)
6b	*Inaccuracy Localization*	(5), (8), (9), (13)

A. *Inter-domain link delay modeling*

For modeling inter-domain link delay components, a training database (DB) with inter-domain link delays ($y_l(t)$) is constructed based on the inter-domain link delays estimation, given intra-domain delay models and the measured throughput and e2e delay for the paths.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

We assume that the intra-domain delay models have been received from the domain controllers and are stored in a DB, and a meaningful phase of monitoring data collection spanning *Ttr* monitoring time periods has been carried out and the data are stored in an e2e monitoring DB (this phase concerns blocks labeled 0a and 0b in Fig. 4); the proper value of *Ttr* needs to be chosen considering the trade-off between the required sample size to obtain meaningful inter-domain link models and the time needed to collect all monitoring measurements.

Once data are available, the delay component estimation starts, and for each collected measurement $<x_p(t), y_p(t)>$, t=1..*Ttr*, several steps are executed to infer the components of such delay introduced for each inter-domain link crossed by a path. First, the domain delay models are used to produce the delay expected ($\varphi_{pd}(\cdot)$) in every domain (block 1). Next, block 2 focuses on isolating the per-path aggregated inter-domain delay component $y_p^{inter}(t)$, defined as the remainder of delay that cannot be explained by the summation of the expected domain contributions predicted by domain models; $y_p^{inter}(t)$ can be formally defined as:

$$y_p^{inter}(t) = y_p(t) - \sum_{d \in D} \varphi_{pd}\left(x_p(t)\right) \tag{2}$$

Consecutively, block 3 processes jointly all $y_p^{inter}(t)$ values to disaggregate the delay per-inter-domain link. The result of this step allows inferring inter-domain link delays $y_l(t)$ from monitoring data. This inference is supported by the assumption that the expectation ($E[\cdot]$) of per-path aggregated inter-domain delay component equals the sum of the expectations of the delays introduced by each inter-domain links of the path, i.e.:

$$E\left[y_p^{inter}(t)\right] = \sum_{l \in L} \delta_{pl} \cdot E[y_l(t)], \quad \forall p \in P \tag{3}$$

According to Eq. (3), the estimation of $y_l(t)$ values given a set of paths P and a set of per-path aggregated inter-domain delays $y_p^{inter}(t)$ can be done by simple regression techniques. In this work, we propose implementing the disaggregation block (3) by using the least absolute deviation regression [29], which entails solving the *Link Delay Disaggregation* optimization problem in Eqs. (4)-(5) independently for each t=1..*Ttr*:

$$\min \sum_{p \in P} \left| y_p^{inter}(t) - \sum_{l \in L} \delta_{pl} \cdot y_l(t) \right| \tag{4}$$

subject to:

$$y_l(t) \in \mathbb{R}^+, \quad \forall l \in \mathrm{L} \tag{5}$$

After solving the above optimization problem, we apply spline smoothing to the obtained $y_l(t)$ values to make them more consistent with the continuous temporal collection and to eliminate those variations resulting from solving each time independently. The results are then used to populate a training dataset (block 4), together with the model input features, i.e., the measurements of the e2e traffic $x_p(t)$.

The resulting dataset can be used for training a fully connected, feedforward Deep Neural Networks (DNN) (block 6a) that predicts φ_l of every inter-domain link as a function of both the traffic $\{x_p(t)\}$ and the route (only inter-domain links) of the paths ($\{\delta_{pl}\}$). The DNN exploits the fact that different paths crossing different inter-domain links could have, however, similar behavior and correlation between traffic and delay. The trained models are stored in a DB and are ready to be used (block 7a).

B. Intra-domain model correction

Although the procedure in the previous subsection has been designed to achieve accurate estimation of the actual inter-domain link delays, there are two cases where that accuracy can be seriously affected: *i*) the availability of a limited number of multidomain paths with few distinct routes can lead to the impossibility of properly isolating and inferring inter-domain link delays. In this regard, our proposed method exploits as much as possible the available information from existing multidomain paths to produce the most accurate compound e2e delay models; *ii*) inaccurate intra-domain delay models. Note that those models are obtained during the commissioning testing phase and updated periodically using active probes, which, as discussed in the introduction, need to be properly configured as otherwise, delay measurements could largely differ from those experienced by the real traffic, thus resulting in inaccurate intra-domain delay modeling.

Especially for the second case, the broker can play a key role in detecting, identifying, and correcting the intra-domain delay model inaccuracies before compound models are used. Note that the benefits are two-fold: 1) after intra-domain models are properly corrected, the broker can make use of accurate compound e2e models without any re-training performed by domains; and 2) the applied corrections can be notified to the affected domain(s), which in turn can use that useful information to tune and adapt its/their mechanism for intra-domain modeling of future services, e.g., using more realistic packet trains used by the active probes.

Before introducing the procedure to detect and identify intra-domain delay model inaccuracies and compute model corrections, the formulation proposed in Section IV.A needs to be revisited.

The presence of inaccuracies in the intra-domain

delay models impacts negatively on the veracity of the assumption formulated in Eq. (2) and now $y_p^{inter}(t)$ values contain not only the aggregated inter-domain link delay component but also the error (underestimation or overestimation) introduced by inaccurate intra-domain delay models. Since inter-domain links can support both accurate and inaccurate paths, finding a common inter-domain link delay value that fits all the paths traversing the link is, by definition, imprecise. In other words, the expression in Eq. (3) defines an expectation of inter-domain link delay that could be far from the true value. Consequently, the condition in Eq. (3) need to be extended to incorporate a *per-path bias* $\beta_p(t)$ that collects those potential intra-domain inaccuracies:

$$E\left[y_p^{inter}(t)\right] = \sum_{l \in L} \delta_{pl} \cdot E[y_l(t)] + \beta_p(t), \quad \forall p \in P \tag{6}$$

Hence, the *Link Delay Disaggregation* optimization problem in Eqs. (4)-(5) needs to be extended to quantify that bias for every path; the mentioned optimization is reformulated as follows:

$$\min \sum_{p \in P} \beta_p(t) \tag{7}$$

subject to:

$$y_p^{inter}(t) = \sum_{l \in L(p)} \delta_{pl} \cdot y_l(t) + u_p(t) - v_p(t), \quad \forall p \in P \tag{8}$$

$$\beta_p(t) = u_p(t) + v_p(t) \tag{9}$$

$$y_l(t) \in \mathbb{R}^+, \quad \forall l \in \mathrm{L} \tag{10}$$

The *Link Delay Disaggregation Bias* optimization problem—that now implements block 3—finds the least absolute deviation of inter-domain link delay components in Eq. (8) with the adjustment of both slack and surplus variables for each path and time t ($u_p(t)$ and $v_p(t)$). Eq. (9) relates per-path bias to slack and surplus variables). Note that as the *Link Delay Disaggregation* problem, the *Link Delay Disaggregation Bias* one needs to be solved for all samples collected during the training period defined by *Ttr*. However, this problem produces not only the set of all inter-domain link delay components but also the set of slack and surplus values to be stored in the inter-domain delay DB (block 4).

Per-path slack and surplus values are analyzed in the inter-domain delay validation (block 5) by solving the *Inaccuracy Detection* problem. This problem aims at identifying the presence of a large bias as a consequence of some intra-domain delay model inaccuracies. Specifically, a decision score s is defined based on key statistical quartiles [30] of the average bias of every path in time. Equation (11) formally describes the computation of the quartiles 25%, 75%, and 100% % of the bias of all paths. The obtained results are then used to compute s in Eq. (12), where the interquartile range $(q_{75\%} - q_{25\%})$ is multiplied by the maximum $q_{100\%}$.

$$\langle q_{25\%}, q_{75\%}, q_{100\%} \rangle = Q\left(\frac{1}{T} \cdot \sum_{t=1..Ttr} \beta_p(t), \ \forall p \in P; \langle 25\%, 75\%, 100\% \rangle\right) \tag{11}$$

$$s = (q_{75\%} - q_{25\%}) \cdot q_{100\%} \tag{12}$$

Intra-domain model inaccuracies increase the bias of some paths, so we expect that both maximum $q_{100\%}$ and interquartile range $(q_{75\%} - q_{25\%})$ increase, which makes that the proposed score sharply increases. In the case that the score is under a predefined threshold, then the inter-domain components in the dataset (block 4) are validated and they can be used for training the DNN (block 6a); otherwise, the phase of inaccuracy localization starts.

C. Inaccuracy localization

Upon the detection of inaccuracy, the localization of the source of such inaccuracy (block 6b) can be done by solving the *Inaccuracy Localization* optimization problem, which is a variation of the *Link Delay Disaggregation Bias* one. This variation requires selecting one domain d at a time and the set of paths crossing it. The formulation of the *Inaccuracy Localization* problem is as follows:

$$\min \beta_d(t) = \frac{1}{|P \backslash P(d)|} \cdot \sum_{p \in P \backslash P(d)} \beta_p(t) \tag{13}$$

subject to: Constraints (5), (8), and (9)

The *Inaccuracy Localization* problem excludes domain d from the objective function and therefore, slack and surplus variables of the paths traversing d can take any value with no additional cost. Then, if the inter-domain link delays can be obtained without significant bias of the non-affected paths, the selected domain is a source of inaccuracy. Therefore, we solve the *Inaccuracy Localization* problem for every domain and select the one with the lowest bias $\beta_d(t)$ as responsible for the inaccuracy.

Finally, block 7b estimates—e.g., by applying cubic spline regression [31]—the needed correction φ^*_{pd} as a function of the load using the obtained slack and surplus values. Such corrections are stored in a DB (block 8b), so the prediction of intra-domain models is computed as the sum of the prediction of the model itself plus the prediction of the correction model.

V. Illustrative Results

This section presents simulation results to validate the blocks, models, and procedures described in Section IV.

Table III. Characteristics of Generated Traffic

	min	max	median	mean	std. dev.
Traffic (Gb/s)	0.78	10	3.68	3.95	2.56
Delay (ms)	9.42	75.38	13.24	19.32	12.92

A. Simulation Scenario

CURSA-SQ [32] was used as a simulation environment. The CURSA-SQ methodology was tuned with the experimental measurements in [12], and successfully used to reproduce realistic packet scenarios, including converged fixed-mobile [33] and time-sensitive networks [34].

A multidomain topology was reproduced, which consisted of four domains and 12 inter-domain unidirectional links connecting border routers of two different domains. Seven customer sites (endpoints of multidomain paths) were connected to domain edge routers in every domain. Five distinct candidate routes ranging between 500 km and 1000 km were considered for every pair edge-border and border-border routers, to emulate a wide variety of intra-domain networks. The topology represents a moderated but realistic size of a multidomain topology [23]. Based on this reference topology, scenarios consisting of several multidomain paths traversing 3 out of 4 domains, were generated. Every multidomain path requested 10 Gb/s maximum capacity and was routed randomly, firstly at inter-domain level (by selecting a sequence of domains and inter-domain links) and secondly at intra-domain level (by selecting a candidate route per each path segment), being the total length of the e2e routes between 1,000 and 3,000 km. Inter-domain links were dimensioned to fit the sum of maximum traffic of all traversing paths. Besides, background single domain 10 Gb/s paths, using the same candidate intra-domain routes, were added to generate different scenarios, where the proportion of delay introduced by inter-domain links with respect to the total e2e delay (hereafter, denoted as proportion ρ) varies. CURSA-SQ was used to generate seven days of traffic, emulating daily variations under the assumption of stationary traffic conditions for each configuration: <number of multidomain paths, ρ>. The generated traffic was injected in the scenario defined above. The characteristics of the traffic are summarized in Table III.

The next subsections present the obtained results.

B. Inter-domain link delay modeling

Let us first study and validate the accuracy of the proposed methodology for inter-domain link delay estimation, i.e., blocks 2 and 3 in Fig. 4 including the *Link Delay Disaggregation* problem defined in Section IV.A, assuming the availability of accurate intra-domain delay models. Since we simulated each delay component to elaborate e2e measurements, we had access to the *real* delay introduced by each inter-domain link. Therefore, we compared the estimated delay against the real one and computed the inter-domain link delay error estimation.

Fig. 5 presents the obtained results as a function of two of the most relevant factors impacting on such error: *i*) the number of multidomain paths; and *ii*) parameter ρ. In particular, Fig. 5a plots the performance for several values of ρ as a function of the number of paths, whereas Fig. 5b highlights the results for 60, 120, and 240 paths as a function of ρ. We observe that the higher the number of multidomain paths is, the more accurate the delay estimation becomes. However, the proportion of multidomain traffic is crucial to determine the magnitude of this error. In scenarios where the intra-domain delay is far greater than the inter-domain one (ρ=15%), achieving the desired target error below 5% is not possible even when many multidomain paths are available. However, as soon as the impact of inter-domain link delay increases, the error sharply drops. Indeed, we observe in Fig. 5b that a small relative error (< 5%) is achieved for ρ=30% with 160 multidomain paths, which is a reasonable configuration in realistic multidomain scenarios [23]. This observation highlights the importance of accurate estimations of the inter-domain link delay components.

Assuming a realistic configuration of 240 paths and ρ=25%, let us now focus on analyzing the goodness of fit of the compound model in Eq. (1). We assume that block 6a in Fig. 4 is executed after one day of e2e traffic and delay measurements being available, i.e., the number of periods *Ttr* of monitoring data collection for model training purposes was set to 1,440 (1 day with 1 min. granularity).

These data were split 80-20% for training and validation, respectively and used to find the configuration of the DNN that returns the best performance in terms of accuracy with a moderated size to avoid overfitting. The backpropagation training algorithm using batches of 64 samples was considered; training took 30 min, which is much less that the time needed to collect used monitoring data. After evaluation of a wide range of number of hidden layers and size, a fully-connected DNN with three hidden layers and 10 neurons per layer implementing the hyperbolic tangent activation function was selected. Fig. 6 shows both training and validation loss, computed as the mean absolute deviation between the predicted and the actual delays as a function of the number of training epochs, for inter-domain links between domains 1 and 2. We observe that the convergence to an accurate and robust model is achieved after only a small number of training epochs; a common behavior observed in the rest of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

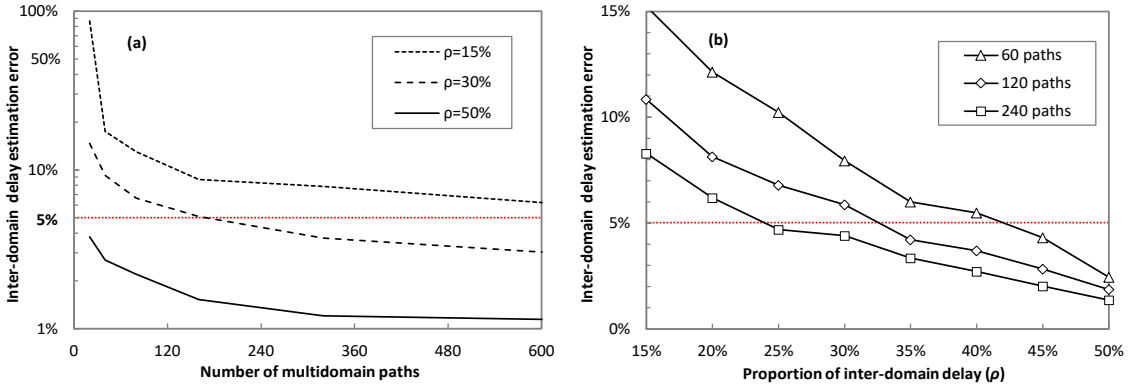

Fig. 5. Inter-domain link delay error estimation vs. number of multidomain paths (a) and proportion ρ (b).

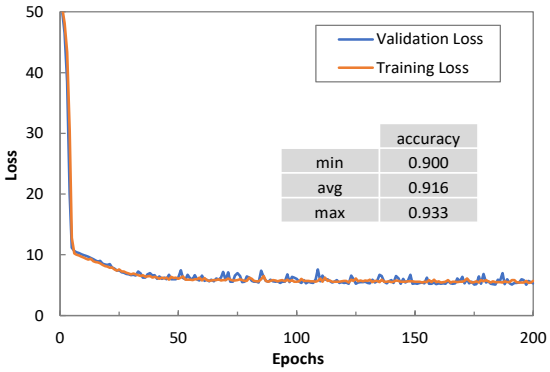

Fig. 6. Domain 1 to domain 2 link modeling performance.

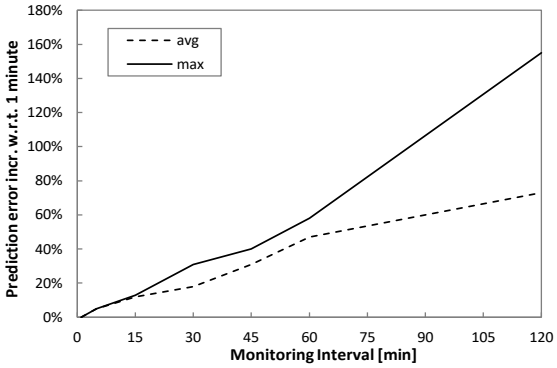

Fig. 7. Increment in prediction error vs. monitoring interval.

inter-domain link models. The embedded table in Fig. 6 summarizes the prediction accuracy for the selected inter-domain link, where accuracy above 0.9 allows validating the reliability and high accuracy of the proposed method to estimate and model inter-domain link delay.

Let us study now how increasing the monitoring period impacts the accuracy of the model. Fig. 7 presents the incremental maximum and the average error when larger monitoring periods are considered. We observe that the 1 min monitoring period can be relaxed to 5 or even 15 min without a major impact on the models' accuracy.

C. *Benchmarking*

Once the accuracy of the compound model has been validated, let us now compare this approach against two different methods that can also be used to compute and predict not only the e2e delay, but also the delay introduced by the domains and inter-domain links that a given path traverses. The methods are based on network tomography [13] and INT [5].

Since no information about the internal topology of the domains is available in the defined multi-operator scenario, we have applied network tomography to infer the delay of domains and inter-domain links from e2e measurements, predictions, and inter-domain routing. Specifically, the delay introduced by every component is computed so to minimize the mean square error between the approximated delay of every path, computed as the sum of the inferred delays of crossed segments, and the real e2e measurement. Regarding INT, the broker would be able to collect per-packet telemetry data, which includes the real delay introduced at every hop from source to destination. To hide internal domain topology, we assume that only edge and border routers add INT measurements to the packets.

For the sake of a fair comparison, we assume that all the models are obtained using the measurements obtained for path under study. Note that in the case of network tomography and INT, this assumption entails that there are not measurements available just after the path is established, whereas in the case of the compound model, domain models are available as measurements are collected during the commissioning testing. The same DNN configuration as for the compound model approach is used for these two approaches. We assumed the aforementioned realistic configuration (240 paths, ρ=25%) and conducted exhaustive evaluation of all the approaches for different values of parameter *Ttr*.

Let us first focus on the difference of the e2e delay prediction accuracy among the approaches for just one single path. Fig. 8 shows the measured and predicted e2e delays as a function of normalized input traffic x for the considered approaches. Fig. 8a plots the prediction before collecting any monitoring sample, i.e., just using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

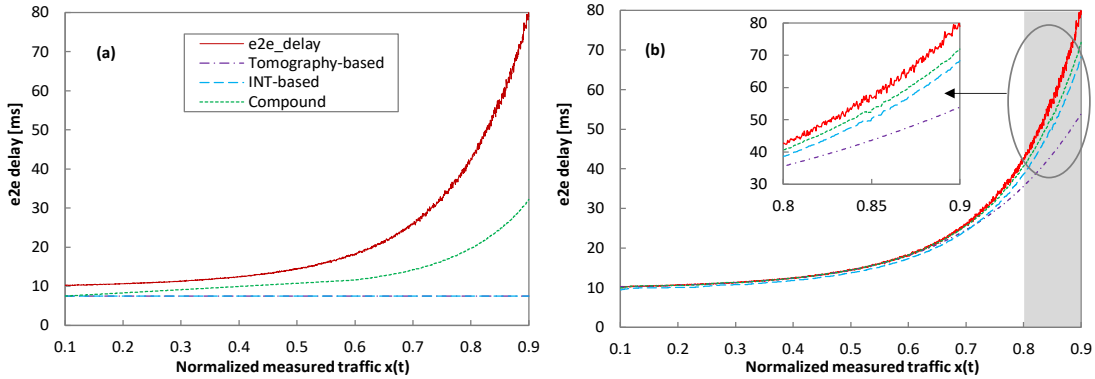

Fig. 8. End-to-end delay prediction example before (a) and after (b) training (Ttr=1440).

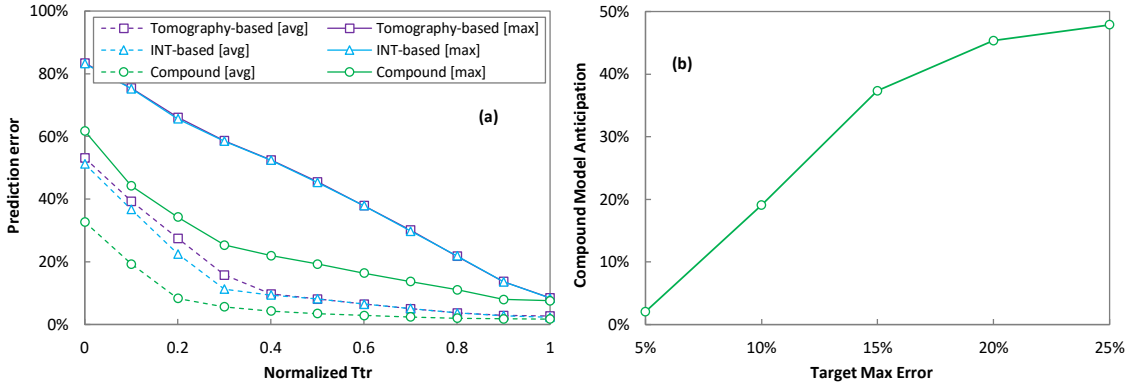

Fig. 9. End-to-end models prediction error (a) and anticipation of compound model w.r.t benchmarking approaches (b).

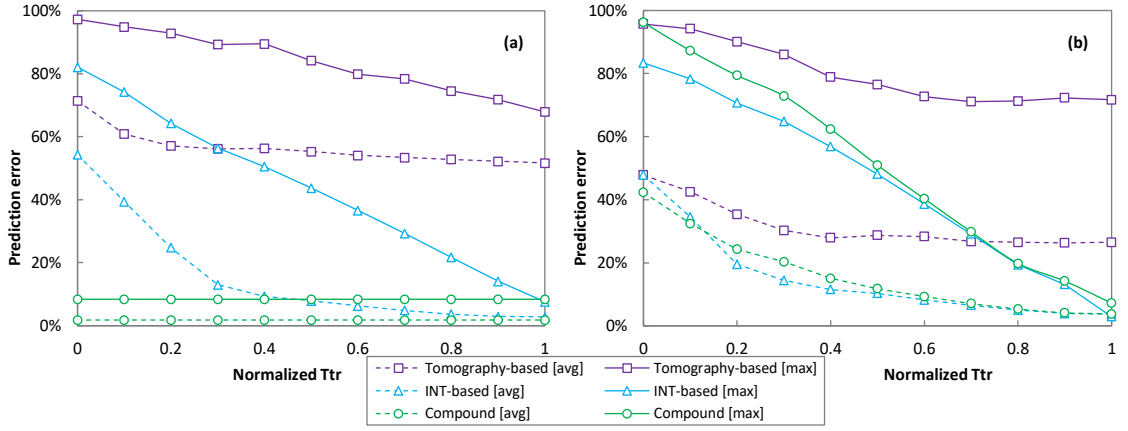

Fig. 10. Domain models prediction error (a) and inter-domain link models prediction error (b).

the available information regarding the path distance in the case of tomography and INT -based approaches, and just with the intra-domain models in the case of the compound model. It is worth noting that the compound model approach offers the possibility to estimate a lower bound of the e2e delay consisting in the sum of all intra-domain delay components. In contrast, that bound, under the other approaches, is simply reduced to the less informative transmission delay. Fig. 8b plots the prediction after data collection and model training for *Ttr*=1440 (i.e., 1 day with 1 min. granularity). In this case, assuming that a wide range of loads was observed during that period, all approaches quickly converge to the measured delay. The compound model and the INT-based approaches closely fit the perfect relation between load and e2e delay, whereas the tomography-based approach still needs some additional monitoring data to better learn the behavior of the e2e delay at high loads.

Fig. 9a shows the relative e2e prediction error (average and max for all 240 paths) as a function of *Ttr*, normalized to the value, where all approaches reached negligible average error (~1%). We observe that the compound model converges faster than other approaches, especially for the maximum error. Assuming that *Ttr* is chosen to guarantee a maximum error under a given target, Fig. 9b shows the relative anticipation of the compound model to achieve such target error w.r.t. the time needed under tomography or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

INT -based approaches (both since they exhibit same maximum error performance in Fig. 9a). Given the results, we can conclude that the compound approach reaches reasonable maximum error (around 10-15%) with a monitoring period between 20% and 35% shorter for training purposes.

Finally, delay prediction accuracy is evaluated in Fig. 10 for all the considered approaches as a function of *Ttr*. The tomography-based fails to produce accurate domain and inter-domain link delay models, as it can be observed in Fig. 10a and Fig. 10b, respectively. The rationale is related to the fact that paths with the same edge/border nodes can follow different inter-domain routes. The compound modeling approach exhibits the best combined performance; on the one hand, domain delay prediction error is constant and remarkably low since accurate models are available from the very beginning of operation, whereas inter-domain link delay prediction accuracy converges rapidly with time, following noticeable close to that given by the models trained with data from INT. In that regard, the INT-based approach produces very accurate models, but needs time to collect the required training dataset.

In conclusion, these results allow to validate the compound approach as a fast and accurate way to obtain e2e delay models and their per-segment components.

D. *Intra-domain model correction*

Let us now focus on validating the models and methods proposed in Section IV.B to detect and localize inaccuracies in intra-domain models during the compound model training phase. Recall that inaccuracy detection is based on solving *Link Delay Disaggregation Bias* optimization problem and computing score *s* in Eq. (11). As score *s* is expected to increase when inaccuracies become larger, to demonstrate the validity of the detection and localization method we need to demonstrate that it is possible to set up a threshold value that discriminates inaccuracies with high precision, thus ensuring that accurate models robustly produce score values clearly under the threshold.

Fig. 11 shows the score in the absence of inaccuracies for all the configurations of the number of paths and ρ already tested in the previous section. We observe that despite some oscillations in the score, the obtained results do not exceed s=1, whose value can be set as the threshold that separates accurate from inaccurate intra-domain models.

Let us now compute the score *s* in the case of inaccurate intra-domain models. To this aim, we consider again the realistic scenario with 240 paths and ρ=25% and we have synthetically generated inaccuracies by adding some additional delay to the intra-domain prediction, thus emulating a hidden delay not considered during the intra-domain model training [12]; inaccuracies range from 1 to 50 ms and were

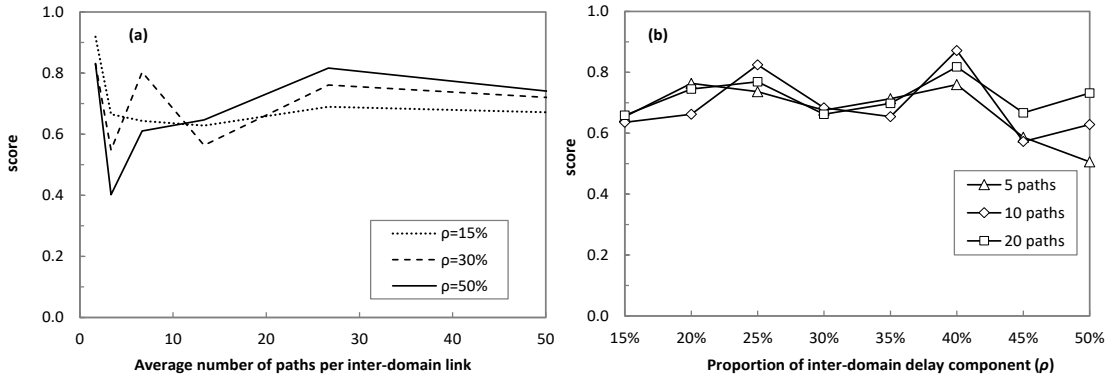

Fig. 11. Score values vs number of paths (a) and proportion ρ (b) in the absence of domain model inaccuracies.

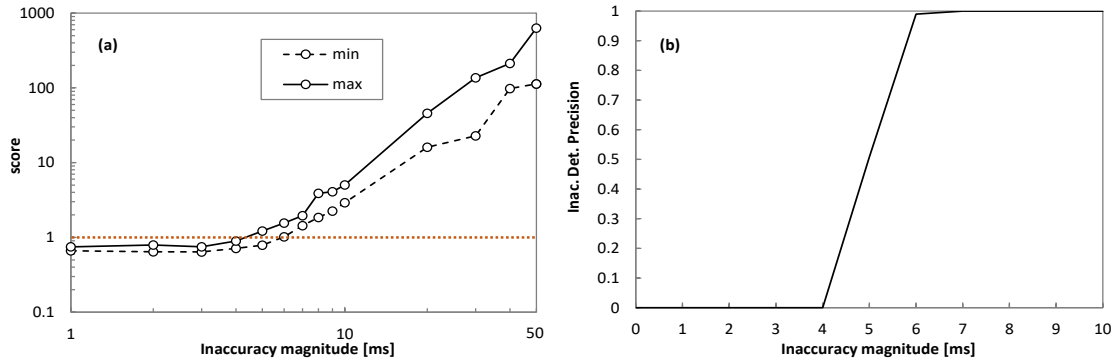

Fig. 12. Inaccuracy detection: score vs inaccuracy magnitude (a) and detection precision vs inaccuracy magnitude (b).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

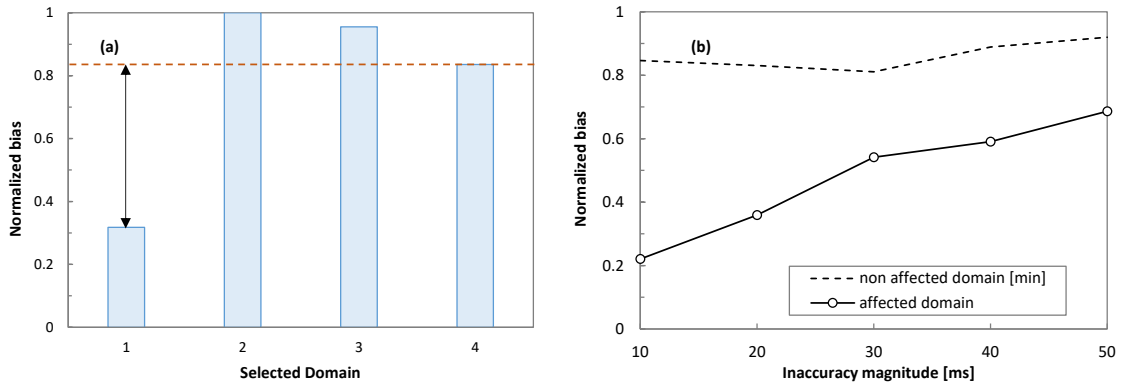

Fig. 13. Inaccuracy localization: example of 10 ms inaccuracy in domain 1 (a) and average results (b).

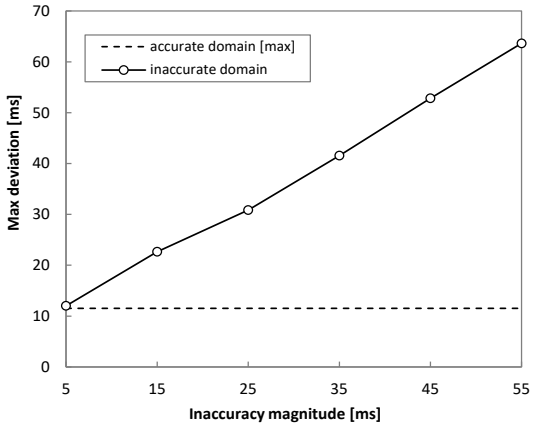

Fig. 14. Inaccuracy detection and localization for sudden inaccuracies.

introduced at one single domain at a time. Fig. 12a shows the score as a function of the inaccuracy magnitude; curves for the minimum and maximum scores obtained for a given path and domain are presented. In the figures, we observe that the threshold defined as s=1 was clearly exceeded when magnitudes over 6-7 ms were introduced. Aiming at providing deeper insight into the performance of accuracy detection, Fig. 12b shows the precision computed as the probability of detecting a true inaccuracy, as a function of the inaccuracy magnitude for the range of magnitudes between 1 and 10 ms. Here, we can confirm 100% of precision for inaccuracy magnitude greater than 7 ms.

E. Inaccuracy localization

Once an inaccuracy has been detected, we need to localize it. Therefore, let us now focus on studying the performance of the inaccuracy localization procedure defined in Section IV.C. Recall that we proposed the *Inaccuracy Localization* optimization problem to that aim, and the domain with the lowest bias β_p is selected as the affected domain.

Fig. 13a shows the results obtained for the scenario used for the detection study, where an inaccuracy of 10 ms was introduced in domain 1; the bias (normalized to the domain that produces the maximum value) for each of the domains is shown. Note that domain 1 clearly presents the lowest bias (around 60% lower than the rest of the domains); the gap between inaccurate and accurate domains is represented by the double arrow in Fig. 13a. Fig. 13b shows the lowest bias (inaccurate domain) and the lowest bias among all accurate domains as a function of inaccuracy magnitude. We observe that the gap is large for all the inaccuracy magnitudes analyzed (from 10-50 ms), which supports a 100% localization precision in the studied range as the bias of accurate domains never drops below the bias of the inaccurate one.

F. Using compound modeling to detect and localize inaccuracies in-operation

The previous studies focused on the training phase; we complement those results with a study of the use of the compound model once in-operation to detect inaccuracies and localize its potential root cause phase.

Let us first focus on analyzing how the compound model can be efficiently used if a sudden event in a domain, e.g., an internal domain reconfiguration, affects the accuracy of the intra-domain delay models for all the multidomain paths crossing the domain. We assume the previous network scenario with 240 paths and consider that the proposed training procedure resulted in accurate compound models; the operation was emulated by generating 60 days of monitoring data samples.

In this case, the inaccuracy is detected by simply comparing the predicted and the measured e2e delays for every multidomain path and analyzing the resulting deviation; a threshold can be configured so its violation triggers its analysis. Fig. 14 shows the average deviation observed for a path crossing the inaccurate domain and for a path routed through other different (and accurate) domains, for different values of inaccuracy magnitude. Thus, setting up a deviation threshold around 15 ms allows us to detect significant inaccuracies above 10 ms. The localization of the inaccurate domain can be implemented by finding the common ones crossed by all affected paths (see [35] for an example applied to single domain networks).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2021.3077736, IEEE Transactions on Network and Service Management

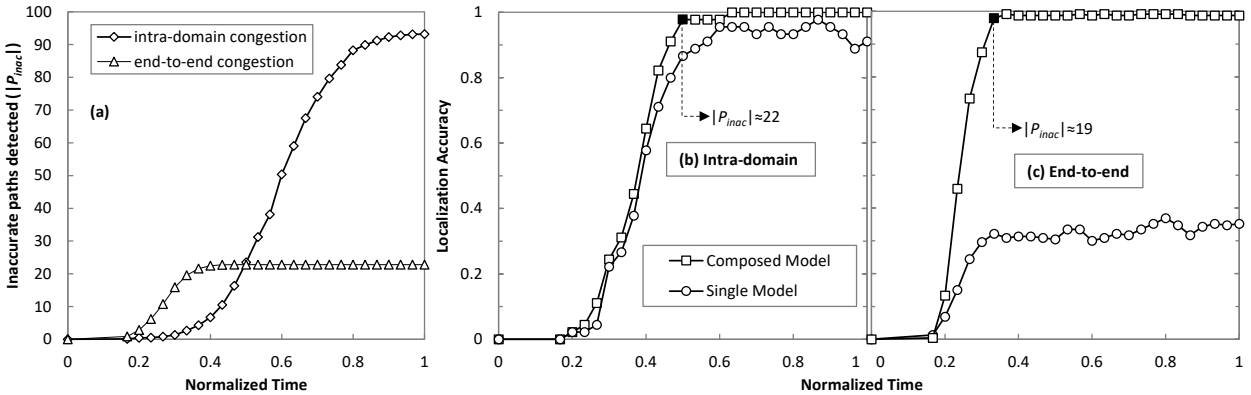

Fig. 15. Inaccuracy detection (a) and localization for gradual inaccuracies (b-c).

Let us assume now a more realistic scenario where not all the paths in a domain are affected by an inaccuracy, its magnitude gradually increases with time, and it affects each path differently. Two different scenarios have been analyzed for a set of affected paths P_{inac}: *i*) *intra-domain increase*, where $|P_{inac}|$ is large; *ii*) *e2e increase*, where $|P_{inac}|$ is a small set and all paths share the same e2e route. To illustrate the difference between both scenarios, Fig. 15a shows the evolution of the number of inaccurate paths detected using the deviation analysis presented in Fig. 14 as a function of the time normalized to the instant when all the paths affected by the inaccuracy are detected. Once a path is detected, every component (domain or inter-domain link) is evaluated as a potential source of inaccuracy. To this aim, a list of *a priori* conditional assumptions for each component, which need to be previously defined, need to be evaluated. In this work, we simplify this list by considering only a maximum delay bound that cannot be exceeded in every domain and inter-domain link. Then, the potential set of inaccuracy sources is updated as soon as a path violates the delay bound until the inaccurate one is isolated.

Fig. 15b-c show the evolution of the localization accuracy in finding the inaccurate model component for the two scenarios considered. For comparison purposes, we have also considered the tomography-based approach where, due to the lack of intermediate accurate model components, localization is done by choosing the domain or inter-domain link supporting the maximum number of inaccurate paths. We observe that the compound model allows almost perfect localization of the inaccurate component regardless of the scenario.

Note that the accuracy of the tomography-based approach is very dependent on the scenario, which discourages from utilizing it for e2e model evaluation purposes. Another result is that the minimum number of paths required to achieve virtually perfect localization accuracy (>99%) remains almost constant in the compound model approach, which is an important outcome as it allows identifying *a priori* condition (number of inaccurate paths detected) that can be applied to decide whether the inaccurate component localization procedure is trustworthy or not and eventually validates the proposed approach based on a compound e2e modeling for different scenarios.

VI. Concluding Remarks

This work proposed a coordination environment for multidomain networks, where domain networks and an inter-domain orchestrator (broker) consistently work for accurate analysis and modeling of e2e delay of multidomain paths. The proposed environment fosters cooperation by distributing tasks between the domains (in charge of modeling intra-domain delay components) and the broker (responsible for modeling inter-domain delay components). As a result of this cooperation, *compound e2e delay models* consisting of the sum of intra- and inter-domain components are obtained and used for multiple purposes, like QoS estimation for connectivity provisioning and reconfiguration upon anticipated QoS degradation.

A numerical evaluation of the proposed compound e2e delay modeling was conducted and compared against reference approaches, where models were trained by using e2e delay monitoring data only (tomography-based) or per-segment monitoring data (INT-based). The results show that: *i*) the inter-domain link delay can be accurately estimated by combining e2e monitoring data and intra-domain model predictions; *ii*) the broker can detect intra-domain model inaccuracies even when their magnitudes are small with respect to the total e2e delay; *iii*) the compound modeling approach converges to highly accurate e2e delay models faster than reference approaches; *iv*) once trained, the compound models can be effectively used to detect sudden in-operation inaccuracies under different potential scenarios, improving the performance of reference e2e delay models. These results validate the proposed cooperative e2e delay modeling architecture and methodology.

REFERENCES

[1] X. Jiang *et al*., "Low-Latency Networking: Where Latency Lurks and How to Tame It," Proceedings of the IEEE, vol. 107, pp 280-306, 2019.

[2] A. Morton, "Active and passive metrics and methods (with hybrid types in-between)," IETF RFC 7799, 2016.

[3] D. Perdices *et al*., "On the Modeling of Multi-Point RTT Passive Measurements for Network Delay Monitoring," IEEE Transactions on Network and Service Management, vol. 16, pp. 1157-1169, 2019.

[4] M. Ruiz *et al*., "Accurate and affordable packet-train testing systems for multi-Gb/s networks," IEEE Comm. Magazine, vol. 54, pp. 80-87, 2016.

[5] L. Tan *et al*. "In-band Network Telemetry: A Survey," Computer Networks, vol. 186, 2021.

[6] L. Velasco *et al*., "Monitoring and Data Analytics for Optical Networking: Benefits, Architectures, and Use Cases," IEEE Network, vol. 33, pp. 100-108, 2019.

[7] L. Velasco *et al*., "An Architecture to Support Autonomic Slice Networking [Invited]," IEEE/OSA J. of Lightwave Tech., vol. 36, pp. 135-141, 2018.

[8] D. Rafique and L. Velasco, "Machine Learning for Optical Network Automation: Overview, Architecture and Applications," IEEE/OSA J. of Optical Comm. and Networking, vol. 10, pp. D126-D143, 2018.

[9] A. Giorgetti, "Proactive H-PCE architecture with BGP-LS update for multidomain elastic optical networks [Invited]," IEEE/OSA Journal of Optical Comm. and Networking, vol. 7, pp. 1-9, 2015.

[10] A. P. Vela *et al*., "Distributing Data Analytics for Efficient Multiple Traffic Anomalies Detection," Elsevier Computer Comm., vol. 107, pp. 1-12, 2017.

[11] M. Mouchet *et al*. "Scalable Monitoring Heuristics for Improving Network Latency," in Proc. NOMS, 2020.

[12] M. Ruiz *et al*., "Modeling and Assessing Connectivity Services Performance in a Sandbox Domain," IEEE/OSA J. of Lightwave Tech., vol. 38, pp. 3180-3189, 2020.

[13] T. He *et al., Network Tomography: Identifiability, Measurement Design, and Network State Inference*, Cambridge University Press, 2021.

[14] S. Nihale *et al*., "Network Traffic Prediction Using Long Short-Term Memory," in Proc. ICESC, 2020.

[15] F. Morales *et al*., "Virtual Network Topology Adaptability based on Data Analytics for Traffic Prediction," IEEE/OSA J. of Optical Comm. and Networking, vol. 9, pp. A35-A45, 2017.

[16] F. Krasniqi *et al*., "End-to-end Delay Prediction Based on Traffic Matrix Sampling," in Proc. IEEE INFOCOM 2020.

[17] K. Rusek *et al*., "RouteNet: Leveraging Graph Neural Networks forNetwork Modeling and Optimization in SDN," IEEE J. on Sel. Areas in Comm., vol. 38, pp. 2260-2270, 2020.

[18] L. Zanzi *et al*., "LACO: A Latency-Driven Network Slicing Orchestration in Beyond-5G Networks," IEEE Trans. on Wireless Comm., vol. 20, pp. 667-682, 2021.

[19] R. Dong *et al*., "Deep Learning for Radio Resource Allocation with Diverse Quality-of-Service Requirements in 5G," IEEE Trans. Wireless Comm., 2020.

[20] A. Solano and L. Contreras, "Information Exchange to Support Multi-Domain Slice Service Provision for 5G/NFV," in Proc. IFIP Networking, pp. 773-778, 2020.

[21] A. Castro *et al*., "Brokered Orchestration for End-to-End Service Provisioning across Heterogeneous Multi-Operator (Multi-AS) Optical Networks," IEEE/OSA J. of Lightwave Tech., vol. 34, pp. 5391-5400, 2016.

[22] M. Ruiz *et al*., "Knowledge Management in Optical Networks: Architecture, Methods and Use Cases," IEEE/OSA J. of Optical Comm. and Networking, vol. 12, pp. A70-A81, 2020.

[23] X. Chen *et al*., "Knowledge-Based Autonomous Service Provisioning in Multi-Domain Elastic Optical Networks," IEEE Comm. Mag., vol. 56, pp. 152-158, 2018.

[24] F. Paolucci *et al*., "Interoperable Multi-Domain Delay-aware Provisioning using Segment Routing Monitoring and BGP-LS Advertisement," in Proc. European Conference on Optical Communication (ECOC), 2016.

[25] X. Chen *et al*., "Multi-Agent Deep Reinforcement Learning in Cognitive Inter-Domain Networking with Multi-Broker Orchestration," in Proc. OFC, 2019.

[26] Z. Zhong *et al*., "Routing without Routing Algorithms: An AI-Based Routing Paradigm for Multi-Domain Optical Networks," in Proc. OFC, 2019.

[27] X. Chen *et al*., "Demonstration of distributed collaborative learning with end-to-end QoT estimation in multi-domain elastic optical networks," OSA Optics Express vol. 27, pp. 35700-35709, 2019.

[28] L. Velasco *et al*., "Designing, Operating and Re-Optimizing Elastic Optical Networks," IEEE/OSA J. of Lightwave Tech., vol. 35, pp. 513-526, 2017.

[29] S. Eakambaram and R. Elangovan, *Least Absolute Deviation Regression Theory and Methods: Monograph on LAD Regression Theory*, LAP Lambert, 2011.

[30] A. Bartolucci, K. Singh, and S. Bae, *Introduction to Statistical Analysis of Laboratory Data*, Wiley, 2015.

[31] G. Knott, *Interpolating cubic splines*, Birkhäuser, 2000.

[32] M. Ruiz *et al*., "CURSA-SQ: A Methodology for Service-Centric Traffic Flow Analysis," IEEE/OSA J. of Optical Comm. and Networking, vol. 10, pp. 773-784, 2018.

[33] A. Bernal *et al*., "Near Real-Time Estimation of End-to-End Performance in Converged Fixed-Mobile Networks," Elsevier Computer Comm., vol. 150, pp. 393-404, 2020.

[34] L. Velasco and M. Ruiz, "Supporting Time-Sensitive and Best-Effort Traffic on a Common Metro Infrastructure," in IEEE Comm. Letters, vol. 24, pp. 1664-1668, 2020.

[35] S. Barzegar *et al*., "Soft-Failure Detection, Localization, Identification, and Severity Prediction by Estimating QoT Model Input Parameters," IEEE Transactions on Network and Service Management, 2021.