Disaggregated Delay Modeling in Multidomain Networks

Davide Careglio*, Marc Ruiz, and Luis Velasco
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail: davide.careglio@upc.edu*

ABSTRACT

Accurate delay estimation is one of the enablers of future network connectivity services. 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 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.

Keywords: cooperative learning; multidomain networks; end-to-end delay.

1. 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 beyond 5G and 6G 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. QoS performance should be thus monitored periodically and *passive* and *active* monitoring techniques have been defined (see [2]). On their side, customers can get monitoring data from the operator only in case in case of single-operator scenarios) or they can install their own active probes and measure e2e QoS. Telemetry measurements collected from the network can be used for the estimation of the performance of packet connections, e.g., by training a Machine Learning (ML) model [3]. 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 (*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.

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 [4]. 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.

In this paper, we summarize our previous work in [5] and focus on inter-domain link modeling, intra-domain model correction, and inaccuracy detection and localization.

2. END-TO-END AND PER-DOMAIN DELAY ESTIMATION

Figure 1 shows the control architecture considered in this paper. We assume a dynamic scenario where a *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 is assisted by a digital twin that uses ML- based models for performance prediction. 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.

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). The embedded graph in Fig. 1 illustrates an example of delay model. After the connection is set up, end-to-end delay ML models in the digital twin 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

979-8-3503-0302-5/23/$31.00 ©2023 IEEE

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) 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.

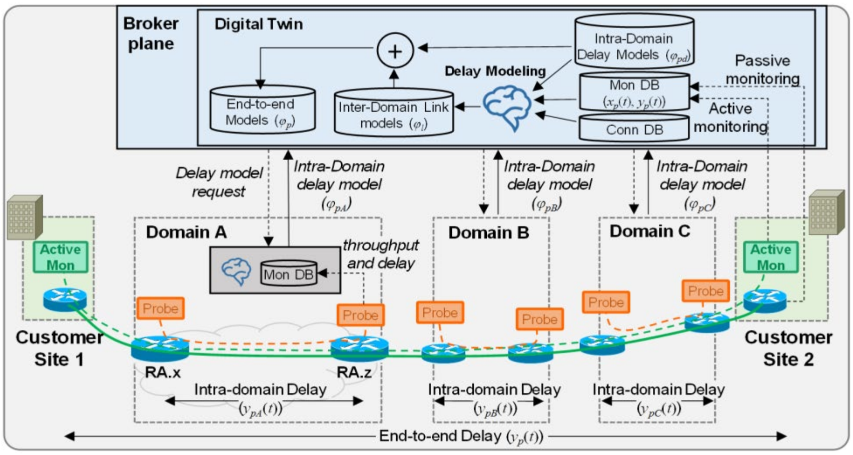
Broker plane
Digital Twin
+
Intra-Domain Delay Models (φ_{pd})
Delay Modeling
Passive monitoring
Mon DB ($x_p(t)$, $y_p(t)$)
Active monitoring
End-to-end Models (φ_p)
Inter-Domain Link models (φ_l)
Conn DB
Delay model request
Intra-Domain delay model (φ_{pA})
Intra-Domain delay model (φ_{pB})
Intra-Domain delay model (φ_{pC})
Domain A
throughput and delay
Mon DB
Domain B
Domain C
Active Mon
Active Mon
Probe
Probe
Probe
Probe
Probe
Probe
Customer Site 1
Customer Site 2
RA.x
RA.z
Intra-domain Delay ($v_{pA}(t)$)
Intra-domain Delay ($v_{pB}(t)$)
Intra-domain Delay ($v_{pC}(t)$)
End-to-end Delay ($y_p(t)$)

Figure 1. Example of e2e delay and control architecture.

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. To break the black-box, 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. Note that intra-domain segment models need to be computed by the domains themselves. 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.

3. COMPOUND E2E DELAY MODELING

This section details the processes that together provide accurate compound e2e delay models. 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 Eq. (1).

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

Figure2 presents the main building blocks and their relationships for compound e2e delay modelling at the broker plane. Blocks have been conveniently numbered to facilitate the ongoing description. As depicted in Fig. 2, the main blocks can be organized into three 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.

Inter-domain link delay modeling: For modeling inter-domain link delay, 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. We assume that the intra-domain

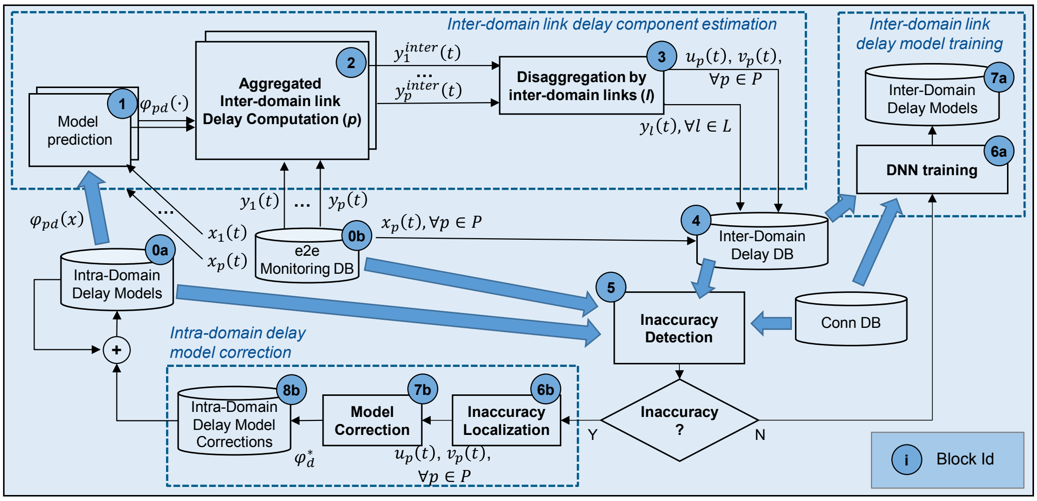
Inter-domain link delay component estimation
Inter-domain link delay model training
Model prediction
1
$\varphi_{pd}(\cdot)$
Aggregated Inter-domain link Delay Computation (p)
2
$y_1^{inter}(t)$
...
$y_p^{inter}(t)$
Disaggregation by inter-domain links (l)
3
$u_p(t)$, $v_p(t)$, $\forall p \in P$
$y_l(t), \forall l \in L$
Inter-Domain Delay Models
7a
DNN training
6a
$\varphi_{pd}(x)$
$y_1(t)$ ... $y_p(t)$
...
$x_1(t)$
$x_p(t)$
$x_p(t), \forall p \in P$
e2e Monitoring DB
0b
Intra-Domain Delay Models
0a
Inter-Domain Delay DB
4
5
Inaccuracy Detection
Conn DB
+
Intra-domain delay model correction
Intra-Domain Delay Model Corrections
8b
φ_d^*
Model Correction
7b
Inaccuracy Localization
6b
$u_p(t)$, $v_p(t)$, $\forall p \in P$
Inaccuracy ?
Y
N
i
Block Id

Figure 2. Main building blocks for training and correcting delay models at the broker plane.

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. 2); 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 the 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 isolates the per-path aggregated inter-domain delay component $y_p^{inter}(t)$, defined as the remainder of delay that cannot be explained by the sum of the expected domain contributions predicted by domain models; $y_p^{inter}(t)$ can be formally defined as Eq. (2).

Algorithm I. Inter-Domain Link Delay Modeling.

```
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)
    $\varphi^*_d$ ← Model_Correction ($d^*$,$\{u_p\}$, $\{v_p\}$) (block 7b)
    Corrections ←Corrections ∪ $\{\varphi^*_d\}$
return $\{\varphi_l\}$, Corrections
```

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

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

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

subject to:

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

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 (Eq. (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, which entails solving the *Link Delay Disaggregation* problem in Eqs. (4)-(5) independently for each t=1..*Ttr*. 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 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 similar behavior and correlation between traffic and delay. The trained models are stored in a DB (block 7a).

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 3 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 (Eq. (6)). Hence, the *Link Delay Disaggregation* problem in Eqs. (4)-(5) needs to be extended to quantify that bias for every path (Eqs. (7)-(10)).

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

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

subject to:

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

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

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

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 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 $q_{100\%}$.

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

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

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 inaccuracy localization starts.

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:

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

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—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.

4. CONCLUSION

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.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Smart Networks and Services Joint Undertaking under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101095890 (PREDICT-6G), from the Spanish MINECO and the European Union though the NextGeneration UNICO5G TIMING (TSI-063000-2021-145/148/149), and the AEI IBON (PID2020-114135RB-I00) projects and from ICREA.

REFERENCES

[1] X. Jiang *et al.*, "Low-latency networking: Where latency lurks and how to tame it," in *Proc. IEEE*, vol. 107, 2019.

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

[3] D. Rafique *et al.*, "ML for opt. network automation: Overview, architecture and applications," *JOCN*, vol. 10, 2018.

[4] A. P. Vela *et al.*, "Distributing data analytics for eff. multiple traffic anomalies det.," *Comp. Comm.*, vol. 107, 2017.

[5] F. Tabataba *et al.*, "Coop. learning for disaggregated delay modeling in multidomain netwks.," *TNSM*, vol. 18, 2021.