PILOT: A Methodology for Modeling the Performance of Packet Connections

Jaume Comellas[*], **Marc Ruiz, and Luis Velasco**
Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail: jaume.comellas@upc.edu*

ABSTRACT

Network Services automation requires predictable Quality of Service (QoS) performance, measured in terms of throughput, delay and jitter, to allow making proactive decisions. QoS is typically guaranteed by overprovisioning capacity dedicated to the packet connection, which increases costs for customers and network operators, especially when the traffic generated by the users and/or the virtual functions highly varies over the time. This paper presents the PILOT methodology for modeling the performance of packet connections during commissioning testing in terms of throughput, delay and jitter. PILOT runs in a sandbox domain and constructs a scenario where an efficient traffic flow simulation environment, based on the CURSA-SQ model, is used to generate large amounts of data for Machine Learning (ML) model training and validation. The simulation scenario is tuned using real measurements of the connection obtained from a set of active probes.

Keywords: sandbox domain, performance modeling and prediction.

1. INTRODUCTION

Connectivity services related to beyond 5G and 6G require not only stringent, but also more predictable quality of service (QoS) performance, measured in terms of key performance indicators (KPI) such as throughput, delay, and delay variation (*jitter*). Solutions currently under research ensure QoS of packet connections by reserving specific network resources at the cost of high overprovisioning. Even though the performance can be bounded, they cannot be precisely estimated, which might be of interest for both network operators and for customers.

The performance of a layer 2 (L2) / layer 3 (L3) connection can be assessed during the commissioning phase through active monitoring, as we demonstrated in [1]. The methodology is different from that proposed by the IP Performance Measurement (IPPM) working group [2] that uses dissimilar measurements for each performance indicator. We measured a packet connection by using an active probe at the source to inject a train of numbered and time-stamped packets; at the other end, another active probe measures throughput, by using the reception times of every packet, and latency and jitter, by comparing the transmission timestamp with the reception time of each packet. Note that this latency measurement requires a common reference clock for the active probes, which is provided by a global positioning system (GPS) receiver to achieve the needed accuracy.

Machine learning (ML) [3] can help to improve the predictability, as well as to assess the performance of connectivity services; ML models (e.g., deep neural networks -DNN) can be trained and used to estimate the performance of end-to-end packet connections. Note that by considering ML models for such estimation, the details of the network are abstracted and thus, they can be shared with other operators or final customers.

However, to obtain accurate models, training and validation procedures need to be carried out, which entails the availability of a large amount of data. To this end, a telemetry system needs to collate measurements from the nodes. Nonetheless, obtaining specific data to model a given connection takes a long time to collect those data and as traffic flows might be highly dynamic, a different approach is required to reduce the time to create the training and testing datasets. For this reason, a sandbox domain, where ML models can be trained with data from the network and from simulation, can be used. In this regard, the behavior of the queues in packet nodes along a packet connection can be studied using realistic and accurate G/G/1/k queues together with realistic input traffic. In this regard, in our previous works in [4], we proposed a methodology named CURSA-SQ to analyze best effort traffic flows by modeling both service traffic and the behavior of the queues, and it was extended for time sensitive traffic in [5].

As CURSA-SQ is able to reproduce the characteristics of a traffic flow, it can be used to generate the large dataset needed for ML training and validation, thus reducing the amount of real measurements that would otherwise be obtained by the active probes, as well as helping to obtain end-to-end ML models of traffic flow. In this paper, we summarized the work in [6].

2. MODELING AND ASSESSING CONNECTION KPIS

In this section, we detail our proposal for modeling and assessing connection performance, based on one-way active network measurements. Given that one of the main aspects of network automation is to guarantee that provisioned connections actually meet the requested performance, active probes are equipped at the packet layer to measure the performance of packet connections. The active probes are installed in every central office (CO) and connected through an interface of a L2 switch in the internal CO network. Being the active probes connected to interfaces configured in trunk mode, the probes can tag the generated Ethernet frames with the desired VLAN ID, selecting the VLAN to be measured. A simple packet connection is represented Fig. 1. The packet connection has resources reserved in Nodes A, B and C, and its performance is measured by the active probe in

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

CO#1 that acts as a sender and that in CO#3 that acts as a receiver. Although the active probes provide really accurate measurements between them, to generate the large training and validation dataset, we propose to use CURSA-SQ to emulate the complete connectivity set-up and to generate useful models that can be used by a digital twin of the network. However, real measurements are strictly needed to tune CURSA-SQ, which includes *additional* delays (e.g., transmission and other processing-related delays, see [7] for a complete delay model) and fine tune of the queues.

The proposed PILOT methodology runs inside a sandbox domain in control plane, and it includes a module to configure the probes to perform the required measurements based on the characteristics of the services provided by the customer using two random variables (inter-arrival burst rate -IBR- and the burst size -BS) (step A in Fig. 1); the resulting measurements are collected and used to tune a network simulator based on CURSA-SQ (B). Once enough data is generated (C), ML models are trained and validated (D) and they can be shared to a digital twin of the network (E), which will use them to estimate performance metrics based on the load (F). Note that the produced ML models provide a way to reproduce the behavior of the connection without revealing the internal routing or other network details, which facilitates being shared with other operators and end customers. An alternative to ML models would be providing abstracted end-to-end performance data, at the cost of moving large volumes of data.

Figure 2a illustrates the network simulation process, which uses the details of the packet connection received from the network controller, which include the route of the connection, the resources actually reserved, the traffic specifications, and the QoS constraints. In the sandbox domain, the PILOT algorithm defines the scenario for CURSA-SQ-based simulation (Fig. 2b). The CURSA-SQ scenario includes a traffic generator (G) in the source for each of the services specified, a sink node (Sk) the destination, dimensioned queues for each output interfaces in the route of the connection, and delay nodes (d) emulating the delay introduced by the links.

Next, PILOT configures the active probes to measure the performance at every destination CO in the packet connection that will be used to tune the CURSA-SQ scenario, e.g., to ensure that any additional delay is included in the real set-up. Once the probes are configured, PILOT generates measurements configurations that include the definition of bursts mimicking the specified mix of services at meaningful values of IBR and BS random variables. Once the results are received from the destination probes (step B), PILOT uses them to tune the simulation scenario and runs CURSA-SQ to generate a large amount of labeled data for ML training and validation (C). The next section describes the PILOT methodology in depth.

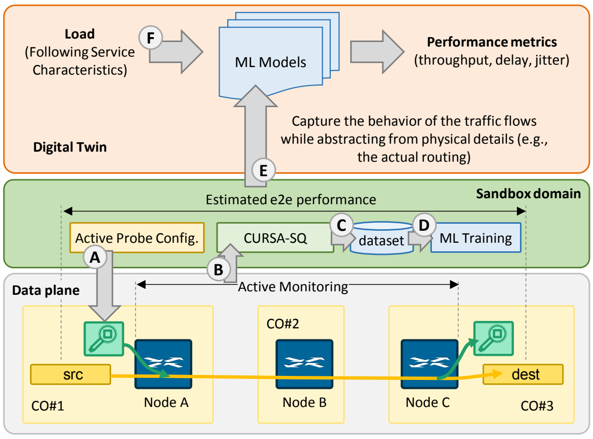
Load
(Following Service Characteristics)
F
ML Models
Performance metrics
(throughput, delay, jitter)
Capture the behavior of the traffic flows while abstracting from physical details (e.g., the actual routing)
Digital Twin
E
Estimated e2e performance
Sandbox domain
Active Probe Config.
CURSA-SQ
C
dataset
D
ML Training
A
B
Active Monitoring
Data plane
CO#2
src
Node A
Node B
Node C
dest
CO#1
CO#3

Figure 1. Sandbox domain and digital twin.

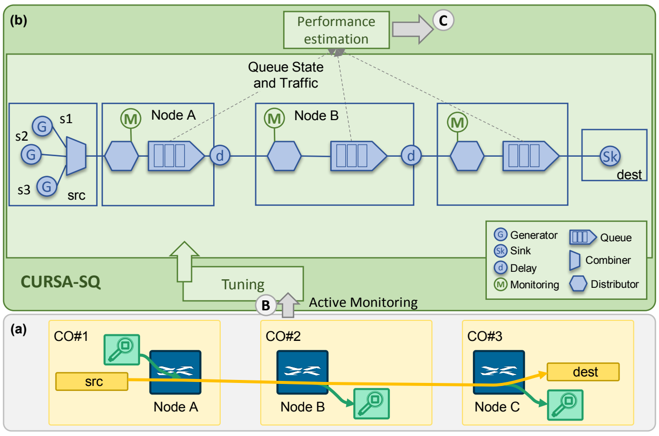

Figure 2. Example of connection (a) and CURSA-SQ simulation (b).

3. COMBINING MEASUREMENTS AND SYNTHETIC DATA

The general scheme of the PILOT methodology is sketched in Fig. 3. PILOT entails three sequential stages to be carried out to produce accurate ML models for each packet connection. PILOT relies on the specification of the traffic mix that the connection will support. Specifically, the mix of traffic is defined in terms of services characterized by, at least, IBR and BS random variables, and a scaling factor. Such specification of the traffic mix is used to generate meaningful active probe configurations in terms of packet trains that are generated by the active probes and which measurements are used to tune the CURSA-SQ scenario. Once experimentally assessed, synthetic data that reproduces the real connection is generated, and accurate ML models can be trained and validated. The following sub-sections elaborate on key PILOT methodology components.

Traffic mix specification

As introduced in the previous section, we assume that a traffic specification is available, which includes the characterization of the mix of services *S* that the connection will contain. Service characterization must include, at least, the statistical distribution and associated parameters of the IBR and BS *burst-level* random variables plus the scaling. From the received service characterization, we define the traffic specification χs for service s, as:

$$\chi_s = \{IBR_s \sim f(\theta_s), BS_s \sim g(\vartheta_s)\}, \forall s \in S \qquad (1)$$

where f and g denote probability distribution functions with their respective parameters. We consider that IBR and BS can be treated as independent variables; indeed, f and g can belong to distinct families of probability distributions.

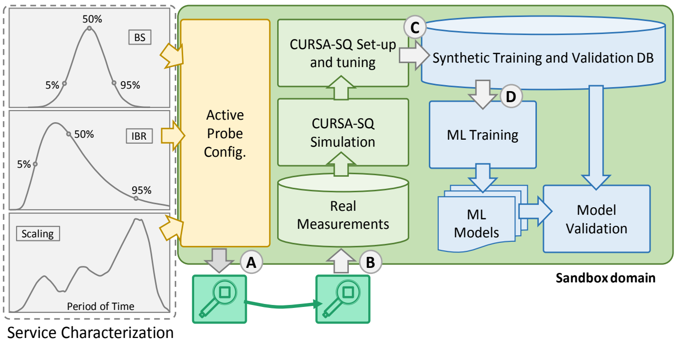

Figure 3. Overview of the PILOT methodology.

In addition, the characterization of the services at *packet-level* can lead to a more precise configuration of active probe measurements. In that regard, *packet size* (PS) is an additional random variable that could be included in χ_S. The expected demand needs also to be included for scaling of each service. We denote this input as $u_s(t)$, where the scaling (e.g., number of individual users) of each service s is defined as a function of time. The specific time range is defined by the customers according to their interests, and it can cover from hours/days/weeks (e.g., a typical tidal profile that periodically repeats on time) to months (e.g., the expected user evolution during connection lifetime). Note that this flexible definition of the expected load opens the possibility to carry out several analysis leading to different KPI modelling for short, medium, and long-term applications. For the sake of simplicity, we assume the same time range for all the services in a connection.

The statistical properties of the services and their expected demand in time are key to understand and define the *traffic flow* $x(t)$ (bitrate, defined in b/s) injected into the connection. For modelling and simulation purposes (mainly for generation), we consider to model services separately, and therefore, the expectation (E) and variance (V) of each service in the flow can be computed as follows:

$$E\big(x_s(t)\big) = u_s(t) \cdot E(BS_S) \cdot E(IBR_S) \qquad (2) \qquad V\big(x_s(t)\big) = u_s(t) \cdot V(BS_S \cdot IBR_S) \qquad (3)$$

where the variance of the product of BS and IBR can be estimated from well-known approximations of the variance of the product of two independent variables [28]. Note that the connection traffic flow $x(t)$ is the aggregation of all $x_s(t)$.

Traffic Sampling and Measurements Configurations

Let us now detail the procedure for sampling the traffic flows $x_s(t)$ to obtain real measurements for those traffic samples using the active probes under realistic traffic conditions (see Fig. 3). We have redefined the synthetic packet generation in the active probes for this purpose, where a measurement request is defined by a number of packet bursts, each containing packets of a given size. The definition of the bursts and the delay between two consecutive ones can be defined to reproduce a desired traffic pattern. The objective is then to define how measurement configurations are created to follow the main statistical characteristics of the specified traffic mix for the connection.

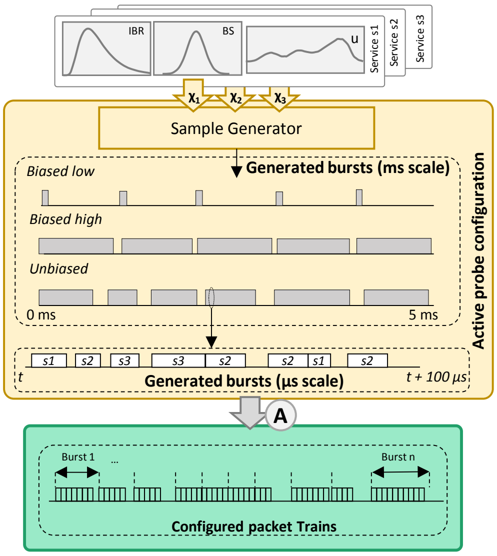

Figure 4. Active probe configuration procedure.

Figure 4 summarizes the concept behind the generation of measurement configurations. Service characteristics are processed by a sample generator module that generates a set of samples of a given duration of bursty traffic; the duration, e.g., 5 ms, is defined by the capacity of the connection. The sample generator module firstly computes the expectation and variance of IBR and BS random variables based on their probability distributions. Then, several time values conveniently spaced in the time range of $u_s(t)$ are selected, thus covering relevant traffic mixes for low, medium, and high loads. For each selected time and mix, samples are generated according the expectation and variance references. In particular, three classes of samples are considered: *i*) *unbiased* samples, where E(IBR) and E(BS) are used for all the services; *ii*) *biased low*, where both E(IBR) and E(BS) are decreased by their respective variance values V(IBR) and V(BS), and *iii*) *biased high*, where both E(IBR) and E(BS) are increased by their respective variance values V(IBR) and V(BS). Regardless of the class, a sample is generated as a sequence of BS and IBR values (mixing services) around their expectation with some additional random variation defined within their variance magnitude. Note that unbiased samples allow measuring KPIs in average cases, which is intended for computing throughput and average latency. On the contrary, biased samples are designed either for measuring additional delays in the absence of queued traffic (*low*) or stressing the connection capacity to compute maximum latency and packet losses (*high*). It is important to analyze the generated samples at different time resolutions. Assuming *self-similarity*, at coarse resolution (ms scale) traffic can be seen as a sequence of on/off periods of mixed services, whereas, at a finer resolution (µs scale), the sequence of on/off periods can be seen between bursts of differentiated services. This degree of detail is required to generate precise active probe configuration.

From the generated samples, a procedure to adjust and configure bursts of trains of packets in the active probe is needed. Thus, a burst in a measurement configuration corresponds to a total (or partial) burst in a sample. Note that, to regularize the length of the bursts of packet trains, a min and max number of packets can be setup.

CURSA-SQ tuning and ML model training

The real measurements for the set of meaningful configurations obtained are used to tune and validate the CURSA-SQ scenario that will be eventually used to generate synthetic data for model training and validation purposes. CURSA-SQ requires configuring a set of traffic generators and this can be done according to equations (2) and (3), as proposed in [5]. However, to reproduce by simulation exactly the same sampled scenarios measured by the active probe, CURSA-SQ traffic generation needs to be altered with the deviation introduced in unbiased measurements.

The generated traffic flows are then propagated through a system of continuous queues Q that models the connection, as represented in Fig. 2. Let us assume that every queue $q \in Q$ is characterized by a unique and common buffer with capacity k (in bytes) and a server rate μ (in b/s). Moreover, $q(t)$ represents the queue state, i.e., the number of bytes in the queue at time t. From such state, partial KPIs are computed for each individual queue. Then, computing connection KPIs, i.e., throughput and latency measurements between the source and all the destinations, is simply the aggregation of partial KPIs computed in queues, as well as in delay nodes.

Despite of the fidelity of the CURSA-SQ-based simulation setup to represent a real connection, there are two main unknowns that need to be discovered after analyzing real measurements. First, the magnitude of the additional delay to be introduced by delay nodes can be easily computed after analyzing the mean latency obtained for biased low measurements. Second, as measurements are correlated to some traffic behavior at the burst-level, but they are actually propagated packet-by-packet during measurements, a mismatch between theoretical and measured traffic behavior can exist. To solve this issue, a *correction factor* (multiplier) can be applied to both expectation and variance configured in the generators in order to fit the characteristics of the expectation and variance measured. A good reference for this purpose is to compare the difference between minimum and maximum latency in the simulation and in the experiments for the case of the biased high monitoring samples. The multiplication factors can be easily obtained to correct the deviation between simulation and experimental values. After this tuning operation, unbiased measurements can be used to validate the accuracy of the simulation environment.

Once CURSA-SQ is tuned and the KPIs obtained by simulation match the experimental ones for the measured samples, a large set of synthetic KPI measurements for a wide range of connection loads along the whole time period can be easily obtained. The data is eventually used for producing ML models that allow a digital twin to estimate KPIs for each connection destination point as a function of the expected traffic mix. In particular, we use DNNs that, given the aforementioned input, return an output vector with estimation of average throughput, average latency, latency bounds, and packet loss (if any).

4. CONCLUSION

A methodology named PILOT has been proposed to provide predictable connectivity services. The PILOT methodology facilitates reducing the cost of overprovisioning. PILOT is based on three main pillars that allow the generation of accurate ML models to estimate the QoS, in terms of throughput, delay and jitter, during commissioning testing: *i*) an efficient traffic flow simulation environment, named CURSA-SQ, to produce large amounts of labeled data for ML training and validation purposes; *ii*) real measurements to tune the CURSA-SQ scenario by discovering additional delay and throughput bottlenecks, which are usually not constant but related to the actual traffic load; and *iii*) specification of the estimated traffic mix that the connection will support are used for data generation, which includes real measurements and traffic generation for the simulation scenario.

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] J. López de Vergara *et al*., "Demo of 100G active measurements in dynamic. provisioned opt. paths," *ECOC*, 2019.
[2] IETF IP Performance Measurement (IPPM) Working Group. [On-line] https://datatracker.ietf.org/wg/ippm/about/.
[3] D. Rafique *et al*., "ML for optical network automation: Overview, arch. and applications," *JOCN*, vol. 10, 2018.
[4] M. Ruiz *et al*., "CURSA-SQ: A methodology for service-centric traffic flow analysis," *JOCN*, vol. 10, 2018.
[5] L. Velasco *et al*., "Supporting TSN and BE traffic on a common metro infrastructure," *Comm. Letters*, vol. 24, 2020.
[6] M. Ruiz *et al*., "Modeling and assessing connectivity services performance in a sandbox domain," *JLT*, vol. 38, 2020.
[7] N. Finn *et al*. (Eds.), "DetNet Bounded Latency," IETF draft work-in-progress, 2022.