# Validation of the CURSA-SQ Methodology

Marc Ruiz* and Luis Velasco

Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.
*Corresponding author: mruiz@ac.upc.edu

***Abstract*—This report collects the results that validate the CURSA-SQ methodology. Models for fixed, mobile, and deterministic networks are compared and validated against reference analytical queue models, through packet-based simulation and experimentally.**

***Keywords*—CURSA-SQ, validation, queue models, packet-based simulation, experimental assessment.**

## I. INTRODUCTION TO CURSA-SQ

The rapid availability of new services makes that network operators cannot exhaustively test their impact on the network or anticipate any capacity exhaustion. This situation will be worse with the imminent introduction of the 5G technology and the kind of totally new services that it will support. In addition, the increasing complexity of the network makes unreachable analyzing its behavior in front of the specific traffic that needs to be supported, which prevents from training human operators and much less, machine learning algorithms that might automatize network operation. In [1], we presented, for the first time, CURSA-SQ, a methodology to analyze the network behavior when the specific traffic that would be generated by groups of service consumers is injected. The methodology allowed to accurately study traffic flows at the input and outputs of complex *fixed networks* with multiples queues systems, as well as other metrics and key performance indicators (KPIs) such as delay, while showing noticeable scalability.

Aiming at supporting a wide range of statistical distributions and functions for traffic characterization and consumers time evolution, CURSA-SQ contains its own modelling approach based on computing approximations of the expectation ($E$) and variance ($V$) of per-user input traffic based on the expectation and variance of two key random variables: the inter-arrival burst rate ($ibr$), defined as the rate of consecutive bursts, and the burst size ($bs$) of services. Note that per-user traffic flows can be conveniently scaled by a large number of users to reproduce aggregated traffic flows. Besides, traffic can be generated with granularity $T$ to simulate a wide range of scenarios, including second and sub-second granularity.

Generated traffic flows are then propagated through the queue system representing the network. Instead of packet-level (discrete) traffic propagation of traditional entity-based queue models and simulators, CURSA-SQ is built on a continuous queue model. Although theoretically continuous queue models scale much better than discrete ones based on packets, they have some additional limitations, such as *i*) the restriction of using infinite queues; *ii*) the impossibility to obtain packet-level measurements such as delay; and *iii*) the impossibility to use practical numerical methods for solving differential equations, such as ordinary differential equation (ODE) methods. To overcome this, CURSA-SQ is built on its own continuous G/G/1/k queue model based on the logistic function that can be easily integrated numerically and extended to consider specific scenarios beyond traditional fixed networks. A comprehensive notation and formulation of the logistic queue model with demonstrations of its main theoretical properties and numerical validation against testing traffic can be found in [2].

In order to extend CURSA-SQ to achieve accurate, scalable and predictive estimation of end-to-end KPIs with sub-second granularity near real-time in converged fixed-mobile networks, in [3] we presented the *mobile network* model that aims at characterizing the Radio Access Network (RAN) segment, where the Evolved Packet Core (EPC) facilitates mobility of User Equipment (UE) and provides Quality of Service (QoS) looking at meeting the needs of services and users with diverse characteristics. According to the general CURSA-SQ methodology, traffic generation can be fairly aggregated in *mobile entities* with similar characteristics, including those related to packet/flow traffic, service, and infrastructure. Under this assumption, a shared medium of capacity $C$ can be modelled as a system of queues, where we define a different queue for each group of UEs consuming the same service. Regarding queues, they are characterized by their capacity and their server rate. The capacity of the queue in an entity by the buffer size typically allocated by the Radio Link Control (RLC) protocol [4] times the number of UEs in the entity. As for the server rate, we redefine that in the original CURSA-SQ model in [2] as being a function of time, which converts the system into a *non-autonomous* ordinary differential equation. Note that server rate is

computed to emulate the shared medium of the cell, while implementing the cell's scheduler (e.g., *proportionally fair* -PF [4]).

Eventually, the development of the fourth industrial revolution (including Industry 4.0) jointly with increasing 5G technology maturity is fostering the need of support for *deterministic networking*. In this regard, several standards from different working groups have been issued focusing on providing bounded QoS in terms of latency (delay), loss, and delay variation, as well as high reliability, e.g. IEEE 802.1 Time-Sensitive Networking (TSN) [5] . Although the initial goal of working groups has been focused on closed environments, interest to extend their scope to provide end-to-end solutions is increasing. End-to-end TSN services entail the support of operators' transport networks that are currently carrying traffic from users, business, and datacenter, just to mention a few on a Best Effort (BE) basis; such traffic is commonly encapsulated into Multiprotocol Label Switching (MPLS) Label-Switched Paths (LSP) at Layer 2 for traffic engineering purposes. In view of the above, in [7] we proposed extensions of the CURSA-SQ methodology to model network interfaces supporting both BE traffic and TSN simultaneously under different TSN standards. In particular, two TSN models relating queue systems and network interfaces were defined based on the IEEE 802.1 standards: the *Synchronous TSN* (*sync*) and the *Asynchronous TSN* (*async*) model. Thus, in the *sync* model, a time window of fixed length (*TW*) is setup and time slices for every TSN flow are reserved, while the rest of the time window that remains unassigned can be used to BE traffic. On the other hand, under the *async* model TSN flows have higher priority than the BE one. Time is processed in small fragments of fixed size, where at every fragment the state of the queues is evaluated and served according to their priority.

In the next sections, we summarize the validation results of all the aforementioned CURSA-SQ models that can be found in [2], [3], [6], and [7]. Table I specifies the validation method used for every model, which includes comparison with other queue models, packet-based simulation, and experimental assessment. Moreover, the details of the implementation and computing resources used in each comparative study are provided.

*TABLE I SUMMARY OF VALIDATION ENVIRONMENT*

| **CURSA-SQ Model** | **SW and HW details** | | | **Validated against** |
|---|---|---|---|---|
| | **Prog. language** | **Server** | **OS** | |
| Fixed Network | Python 2.7 | Intel i7-4790K with 16 GB RAM | Ubuntu 16.04 | • Python 2.7 packet simulator<br>• Experimental assessment |
| Mobile Network | Matlab 2018b | i7-8700 with 16GB RAM | Ubuntu 18.04 | • *ns-3* packet simulator [14] |
| Deterministic Network | Matlab 2018b | i7-8700 with 16GB RAM | Ubuntu 18.04 | • Vacation queue model [17] |

## II. VALIDATION OF FIXED NETWORK MODEL AGAINST PACKET-BASED SIMULATION

For the subsequent studies, we will consider three different services, namely: *VoD*, *Gaming*, and *Internet*. According to the CURSA-SQ methodology in [1], relevant studies available in the literature providing consumer-related and service-related random variables characterization were used to characterize traffic sourced by consumer groups. Table II summarizes the expectation and variance of *ibr* and *bs* for all these services.

*TABLE II SERVICES TRAFFIC CHARACTERISTICS*

| **Service** | ***E*(*ibr*) ($s^{-1}$)** | ***V*(*ibr*) ($s^{-1}$)** | ***E*(*bs*) (MB)** | ***V*(*bs*) (MB)** |
|---|---|---|---|---|
| **VoD** | 0.25 | 2.54e-5 | 3.84 | 1.21 |
| **Gaming** | 1.33 | 0.19 | 0.14 | 0.02 |
| **Internet** | 1.66 | 0.40 | 0.12 | 0.04 |

Let us detail the characterization of the VoD service. Regarding consumer behavior, according to the study presented in [8], the idle time y that an active user spends (e.g., deciding which content to watch) follows the power law probability distribution $p=\alpha \times y^{-\beta}$, with parameters $\alpha=0.43$ and $\beta=1.2$. On the other hand, the duration of the content selected by a user approximates an exponential distribution with a typical mean around 30 minutes and a reasonable maximum of 4 hours [9]. However, users usually stop a reproduction before its completion time. Completion rate depends on the content duration; the longer the duration is, the smaller the completion rate. A Weibull distribution with scale and shape parameters around 75 and 0.8 fits with a large variety of contents' duration [8]. Regarding service-related VoD characteristics, we adopt a typical on-off pattern consisting of an initial 10-20 sec transmission of media contents, followed by a number of 2 sec media segments, until the reproduction finishes

[10]. According to the previously defined statistical distributions, we simulated the activity of a single consumer and stored the time stamp and size of 10.000 traffic bursts. The analysis of this data lead to the VoD consumers traffic characteristics detailed in Table II, that indicates long spaced bursts of large number of bytes.

A similar procedure was followed to characterize gaming and Internet consumers' traffic from key statistical distributions detailed in [11]-[13]. The resultant traffic characteristics differ from that of VoD in both, the frequency of bursts (high *ibr*) and its size (small *bs*). Note that Internet traffic is the one that shows the highest variance in terms of *ibr*, which translates into a less regular traffic pattern.

Aiming at validating the CURSA-SQ methodology including the aggregated input traffic flow model and the logistic queue model, we developed a *packet-based* simulation environment for benchmarking purposes. Specifically, a packet input traffic generator produces packets streams creating of a fixed size creating 1500-byte Ethernet frames, according to the specific mean and variance of *ibr* and *bs*; a packets stream is generated independently for each individual user. Then, the aggregated packets stream is sent to a simple queue system with one *discrete* queue, which processes packet by packet. This combination of packet-based traffic generation and discrete queue simulation provides the baseline performance for comparison purposes.

The CURSA-SQ methodology and the discrete simulator were implemented in Python 2.7 and executed in an Intel i7-4790K -based computer with 16 GB RAM running Ubuntu 16.04.3 LTS. The CURSA-SQ continuous logistic queue was solved by means of ODE solver implementing the Dormand-Prince method [16].

For each defined service, we considered a scenario with a single consumer group configured with a constant number of users. For the sake of a fair comparative analysis, we run several executions with incremental number of users. Every execution generated a random flow of one day long and $T$ = 1sec. that was used for input flow comparison purposes. Then, a sub-second flow with $T$ = 50ms was generated to evaluate the performance of the logistic queue model; both discrete and logistic queues were configured with a 10 Gb/s server.

Fig. 1 shows the average bitrate of the traffic flows of each consumer group against the number of users, using flow-based and packet-based generation. As shown, flow-based generation accurately matches the correlation between generated bitrate and number of users that packet-based generation produced. A detailed accuracy analysis is presented in Table III, where mean and maximum errors of flow-based generation w.r.t. packet-based generation are detailed for every service and different number of users. Mean errors are not higher that 6%, whereas maximum error remarkably decreases with the number of users, reaching no more that 15% in the worst case (for the VoD service) when 100 or more users are considered. Note that gaming and Internet services experience maximum errors not higher that 15% even with 10 users. In light of these results, the accuracy of the proposed statistical methodology to generate aggregated input flows is validated assuming scenarios with a medium/high number of consumers per group.

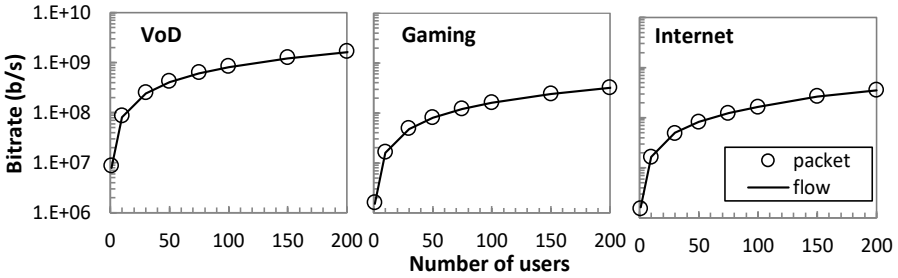


Fig. 1. Input traffic vs. users (reproduced from [1])

*TABLE III RELATIVE ERRORS OF AGGREGATED TRAFFIC FLOWS (reproduced from [1])*

| | **VoD** | | **Gaming** | | **Internet** | |
|---|---|---|---|---|---|---|
| **users** | *mean* | *max* | *mean* | *max* | *mean* | *max* |
| 10 | 6% | 57% | 4% | 14% | 4% | 15% |
| 50 | 5% | 34% | 2% | 5% | 3% | 4% |
| 100 | 4% | 15% | 2% | 2% | 2% | 3% |
| 200 | 4% | 10% | 1% | 1% | 2% | 2% |

A comparison between discrete and logistic queues is shown in Fig. 2a for VoD traffic. In Fig. 2a, the maximum queued traffic is plot as a function of the traffic intensity, computed as the quotient between the average of the aggregated input flow and the speed of the queue server. Note that when the traffic intensity is under about 0.15 the logistic queue is unable to reproduce the behavior of the discrete queue, as for low traffic intensities the discrete behavior becomes more dominant. However, for the scenarios of interest entailing a meaningful traffic intensity, queued traffic evolves similar in both cases, which entails a key numerical evidence to validate not only the logistic queue model but also the procedure to generate aggregated input flows with sub-second granularity.

Looking at analyzing the scalability of both packet-based and flow-based approaches, Fig. 2b presents the total execution time (input flow generation plus queue simulation) as a function of the number of consumers aggregated in the flow. For illustrative purposes, execution time is presented relative to the simulated time, so a value equal to 1 entails simulating the same amount of time that is needed for running the simulation (e.g., 1 day of simulation takes 1 day of execution). As it can be observed, CURSA-SQ runs in few seconds independently of the number of users; this contrasts with the packet-based approach, which execution time is dependent on the number of users and few orders of magnitude larger than that of CURSA-SQ. In addition, the packet-based approach is not practical when a large number of users need to be considered, as its execution time exceeds the simulated time.

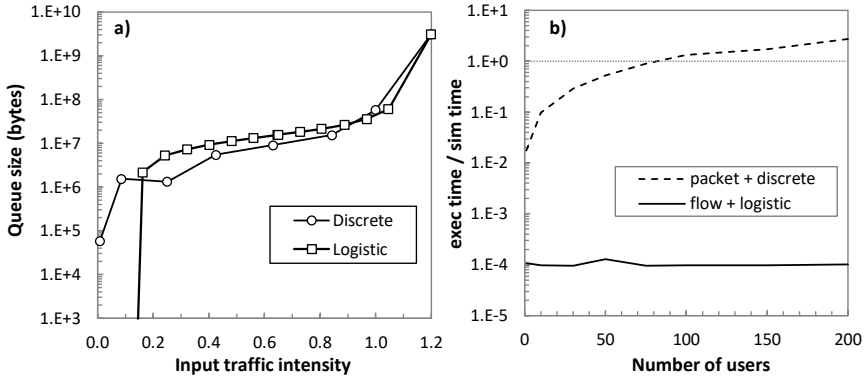


Fig. 2. Queue size (a) and scalability (b) analysis (reproduced from [1])

## III. Experimental Validation of Fixed Network Model

In this section, we show experimental results to validate CURSA-SQ for fixed networks. To this aim, a testbed consisting in two Alcatel-Lucent routers connected through a 10 Gb/s optical link (thus creating a virtual link at the packet layer) was configured. By means of programmable active probes able to send packet train samples following a desired traffic distribution [3], we injected traffic mixing the services already introduced in Section II, i.e. *VoD*, *Gaming*, and *Internet* services. Such samples were generated in the range of normalized load [0.1, 0.95], defined as the average traffic volume over the connection capacity (i.e., 10 Gb/s). Although samples are generated according to the expectation *E* and variance *V* of both *ibr* and *bs* in Table II, two types of samples 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*).

The results in Fig. 3 show the experimental measurements and the simulation data for two different configurations of the CURSA-SQ scenario, focusing on throughput and latency (both average and maximum) KPIs analysis. Precisely, Fig. 3a and Fig. 3b show the results obtained from unbiased samples for average analysis, whereas Fig. 11c focuses on biased high ones, which are relevant to illustrate the real behavior of maximum latency. In the first CURSA-SQ configuration, simulations have been conducted before tuning CURSA-SQ with the real measurements, i.e. according to the network equipment specifications and known service characteristics. The results show a slight reduction of throughput estimation, whereas latency is clearly underestimated due to: *i*) the lack of additional delays consideration, and *ii*) a different latency slope for high loads (clearly visible at loads 0.85 and 0.9). In the second CURSA-SQ configuration, simulations were conducted after tuning CURSA-SQ using biased low monitoring samples at normalized load 0.1 and biased high ones at normalized load 0.9, for additional delay computation and generators corrections, respectively (details of CURSA-SQ tuning procedure from experimental measurements can be found in [3]). As it can be observed, the evolution of all three KPIs as a function of the load closely matches with the experimental measurements.

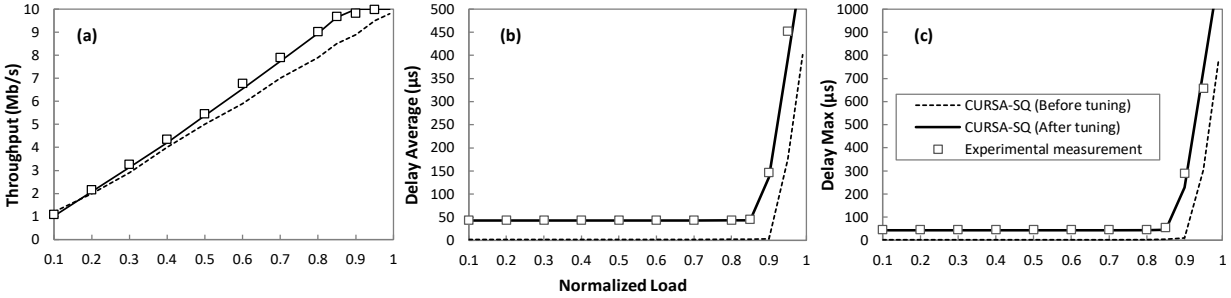


Fig. 3. Experimental and simulation results for KPI estimation: a) throughput, b) average, and c) maximum latency. (reproduced from [6])

## IV. Validation of mobile network Model against packet-based simulation

To validate CURSA-SQ extension for shared medium, we run a number of simulations to compare the performance of our Matlab implementation against that of the ns-3 network simulator implementing the LTE module [14] modeling the full LTE Radio Protocol and the EPC, including the core network interfaces, protocols and entities. Both CURSA-SQ and ns-3 run on an i7-8700 server with 16GB RAM and Ubuntu 18.04.

For this comparative study, a scenario with one single cell was simulated consisting of a base station with a three-sectored antenna, an EPC, and several UEs receiving the traffic (UDP packets) injected by a random bursty traffic generator. Each sector was modelled as a parabolic antenna with a 3dB beam width of 70 degrees and a maximal attenuation of 20dB. For the sake of simplicity, we considered interference-free radio links with line of sight between the base station and the UEs. The ns-3 scenario was configured with the PF scheduler, 1ms transmission time interval, and 5MHz downlink bandwidth. According to the adaptive modulation and coding model in [15], the simulator finds the best *modulation and coding scheme* for a given channel condition. In addition, packet traces generated and used in ns-3 simulation were aggregated in flows with a granularity of 250 ms and used in the CURSA-SQ simulation.

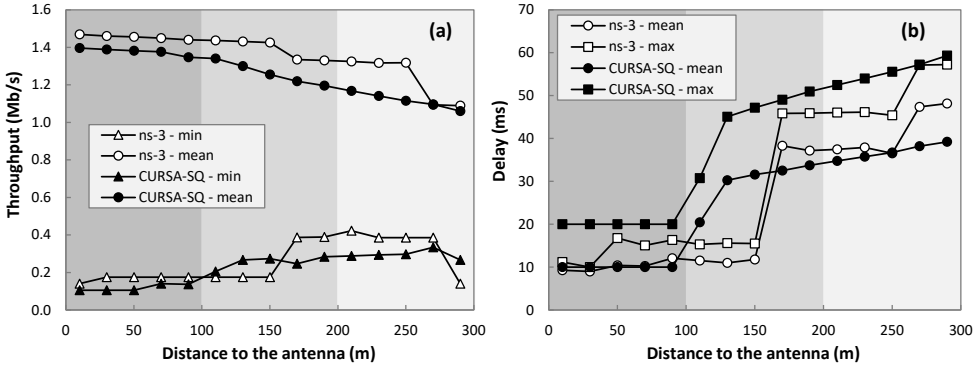


Fig. 4 Throughput (a) and latency (b) in the radio segment vs distance to the antenna. (reproduced from [3])

Table IV Relative difference between CURSA-SQ and ns-3 (reproduced from [3])

| | **Peak-average ratio** | | |
|---|---|---|---|
| | **[1, 1.2)** | **[1.2, 1.4)** | **[1.4, 1.7)** |
| Throughput–min | 3.5% | 6.8% | 10.0% |
| Throughput–mean | 1.5% | 3.7% | 5.3% |
| Delay–mean | 13.2% | 14.1% | 17.4% |
| Delay–max | 20.5% | 15.3% | 12.9% |

Fig. 4 shows the results of the simulation of 15 UEs located between 10 and 290 m from the antenna; CURSA-SQ was configured with one single UE per entity, i.e., 15 entities. The obtained minimum and average throughput and the average and maximum delay per-UE are presented in Fig. 4a and Fig. 4b, respectively, where similar values for both simulation environments can be observed. The larger deviations are for the estimation of the delay for medium

distances (100-200m), where CURSA-SQ overestimates the delay as a consequence of the intrinsic nature of the continuous queue model. However, the impact of such overestimation is minor as they could lead to conservative decisions for the performance analysis module.

Table IV provides an extended comparison of the previous results in terms of the relative difference for quantifying relevant KPIs. A number of repetitions with different random traffic traces and mobility patterns were simulated. The results in Table IV are segmented by different peak/average traffic ratios of the traces; the higher ratio the more bursty the injected traffic. Note that throughput errors typically remain below 10%, whereas higher delay estimation errors are caused by the CURSA-SQ overestimation illustrated in Fig. 4.

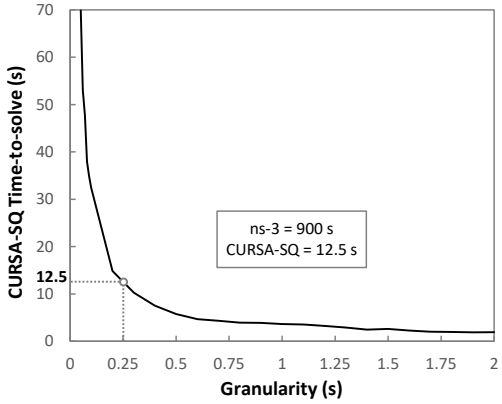


Fig. 5 Time-to-solve vs granularity (reproduced from [3])

Let us now evaluate CURSA-SQ in terms of scalability and its applicability for near real-time KPI estimation. To this aim, let us consider that, in order to make and implement operational decisions, simulations of 2-minute time windows need to be carried out. Fig. 5 shows the time-to-solve one-entity queue system model as a function of the granularity configured in CURSA-SQ. The impact of reducing the granularity is two-fold: while the precision of KPI estimation and the amount of information for performance analysis and decision-making increases, the time-to-solve also increases, which can impact negatively for near real-time operation. As it can be observed, sub-second granularities can be achieved with low time-to-solve times. Specifically, by selecting 250ms granularity, just 12.5 seconds were needed; this is remarkably lower than the simulated 2-minute time-window (10% of the simulated time), which enables its use for near real-time operation. Note that the ns-3 simulation required ~15 min, i.e., 7.5 times the simulated time.

Once the extension of CURSA-SQ for shared medium has been validated in the mobile network segment to compute KPIs assuming a perfect configuration of the system, let us now evaluate the case where input traffic is not accurately forecasted, which could introduce some error in the KPI estimation. To this aim, we conduct a sensitivity analysis that consider errors in the entities' configuration and traffic estimation. Without loss of generality, we focus on the impact of dynamic configuration errors in the estimation of the KPIs on the mobile segment using the configuration of the cell and UEs from the previous study. For the sake of clarity, we analyze both type of errors separately.

Let us first concentrate on analyzing traffic estimation errors. To this end, we configured 1 entity per UE and assumed the traffic flow traces generated from the ns-3 simulator as the real traffic, while added an unbiased Gaussian error to emulate an overall prediction error from the disaggregated traffic estimation and traffic projection submodules. Fig. 6a presents the evolution of the error of the estimation for the minimum throughput and the maximum delay as a function of the normalized error introduced. A close-to-linear relation is observed between the error introduced in the traffic estimation and the error introduced in KPIs estimation when the former is below 30%; above that value, traffic estimation is too poor to be used for KPI estimation. Note that the 30% limit is not stringent, as it is expected that the traffic estimation can remain largely below this threshold.

Let us now focus on evaluating the impact of error in entities configuration, assuming perfect traffic projection. Assuming that 1 entity per UE (15 entities) is the optimal configuration for the current scenario, we introduced error by reducing the number of entities thus, aggregating several UEs per entity. We computed a relative error as the

ratio between the number of entities configured over the optimum one. Fig. 6b plots minimum throughput and maximum delay errors as a function of the error introduced by an inaccurate entities' configuration. A linear evolution of KPI estimation error is again observed with configuration error below 40%, which leads to similar conclusions than those for the traffic estimation error.

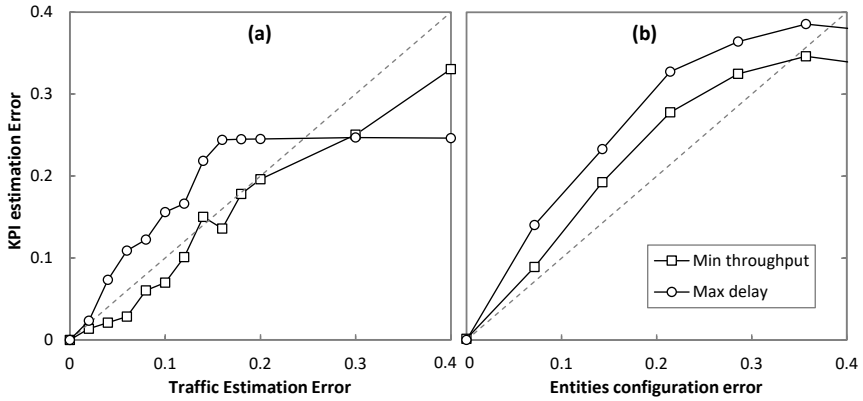


Fig. 6 Error in throughput and delay estimation vs. error in traffic estimation (a) and entities configuration error (b). (reproduced from [3])

## V. Validation of Deterministic network Model against vacation queue model

For this eventual study, we have reproduced a scenario where TSN traffic and BE traffic arrive to a node interface where they are combined. We assume that TSN and BE flows are forwarded through a single 100 Gb/s interface and the sync TSN model was dimensioned with a time window of length T = 125 µs.

TSN and BE traffic mix scenarios can be modeled using the queuing theory with server variations. In this regard, a large number of works can be found in the literature considering server vacations and time-dependent breakdowns (see, e.g., [17]) that limit the availability of the server. Such models can be used to derive expressions for queue system analysis, e.g., to estimate the mean processing time $g$ of a single entity in the queue. In fact, the fraction of time that the server is available ($\rho$) is the main parameter that characterizes the behavior of the queue; from [17], $g \approx 1/(\rho \cdot \mu)$, where $\mu$ is the maximum server rate. We can apply the previous equation for BE traffic under *sync* and *async* approaches just modeling $\rho$. Under the *sync* approach, $\rho_{sync}$ can be fairly computed as the proportion of the remaining time within time window $T$ available for BE traffic, i.e., before allocating reserved slices for TSN flows and the guard band. In the case of the *async* approach, $\rho_{async}$, can be approximated to the proportion of the remaining interface capacity after subtracting the aggregation of the TSN traffic flows.

To compare the results obtained by CURSA-SQ model with the estimation of $g$ from the previous approximated model, we carried out a numerical evaluation assuming 500 100 Mb/s TSN flows (50 Gb/s aggregated TSN traffic) and 0.2 µs time slices under the sync TSN model, so $\rho_{sync}$ = (125-100) / 125 = 0.2. For the async TSN model, $\rho_{async}$ = (100-50) / 100 = 0.50. To relax any assumption on the characteristics and statistical distribution of the incoming BE traffic, which could reduce the applicability of the proposed approximation, we loaded the BE traffic queue with a number $k$ of bytes and solved CURSA-SQ model to compute the time needed to process all queued BE traffic. Calculations were repeated for a wide range of $k$ values.

Fig. 7 shows the obtained results for the BE traffic, where it can be observed that both CURSA-SQ and the approximation model provide very similar values under *sync* and *async* TSN models. In light of this, we conclude that the proposed extensions to the CURSA-SQ continuous model are accurate and can be used for the analysis of the BE traffic performance.

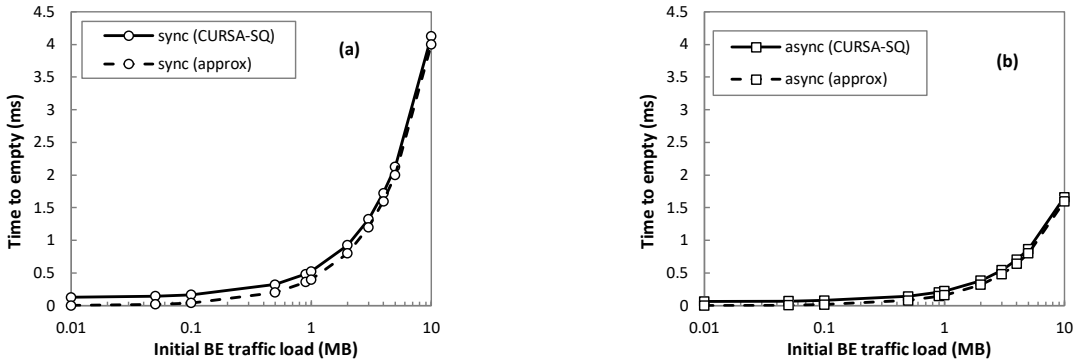


Fig. 7. Accuracy of CURSA-SQ extensions for deterministic networks (reproduced from [7])

## VI. Concluding Remarks

In this report, we presented validation results for all the network models developed under the CURSA-SQ methodology framework. Starting for the fixed network model, the presented results aim at concluding that proposed CURSA-SQ methodology leads to similar results in terms of flow characteristics and queue behavior that the classical packet-based flow generation and discrete queue simulation, and with excellent scalability. In consequence, this methodology can be used to generate traffic for network analysis purposes in complex scenarios. Moreover, experimental validation against a real test-bed showed that relevant KPIs analysis such as throughput and delay as a function of the load based on CURSA-SQ simulation closely matches with the experimental measurements.

In order to validate the mobile network model, CURSA-SQ was validated against the ns-3 network simulator for a pure mobile network scenario. CURSA-SQ estimations probed to be accurate, while running simulation of 120s with granularity 250ms in just 12.5s. These results highlight the outstanding scalability of the proposed method for near real-time computation. In addition, KPI estimation based on CURSA-SQ and predictive models for traffic forecasting allows also detecting errors in traffic prediction, which adds robustness to near-real time KPI estimation.

Last but not least, it has been proven that deterministic networks can be efficiently simulated by means of CURSA-SQ models that closely fits with reference vacation queue models. In this way, complex end-to-end network scenarios where TSN and BE traffic flows share the same network infrastructure can be reproduced.

## References

[1] M. Ruiz, F. Coltraro, and L. Velasco, “CURSA-SQ: A Methodology for Service-Centric Traffic Flow Analysis,” IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, pp. 773-784, 2018.

[2] F. Coltraro, M. Ruiz, and L. Velasco, “Queuing Systems. The Logistic Queue Model”, UPC-DAC-RR-GEN-2020-1, 2020.

[3] A. Bernal, M. Richart, M. Ruiz, A. Castro, and L. Velasco, "Near Real-Time Estimation of End-to-End Performance in Converged Fixed-Mobile Networks," Elsevier Computer Communications, vol. 150, pp. 393-404, 2020.

[4] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution - from theory to practice*, Wiley, 2009.

[5] IEEE Time-Sensitive Networking Task Group. [On-line] http://www.ieee802.org/1/pages/tsn.html.

[6] M. Ruiz, M. Ruiz, F. Tabatabaeimehr, Ll. Gifre, S. López-Buedo, J. López de Vergara, O. González, and L. Velasco, "Modeling and Assessing Connectivity Services Performance in a Sandbox Domain," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 38, pp. 3180-3189, 2020.

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

[8] L. Huang, B. Ding, Y. Xu, and Y. Zhou, “Analysis of User Behavior in a Large-Scale VoD System,” in Proc. NOSSDAV, 2017.

[9] Y. Choi, J. Silvester, and H. Kim, “Analyzing and Modeling Workload Characteristics in a Multiservice IP Network,” in IEEE Internet Computing, vol. 15, pp. 35-42, 2011.

[10] A. Rao et al, “Network characteristics of video streaming traffic,” in Proc. CoNEXT, 2011.

[11] W. Feng et al, “A Traffic Characterization of Popular On-Line Games,” IEEE/ACM Transactions on Networking, vol. 13, pp. 488-500, 2005.

[12] D. Drajic et al., “Traffic generation application for simulating online games and M2M applications via wireless networks,” in Proc. WONS, pp. 167-174, 2012.

[13] X. Wu et al., “Packet size distribution of typical Internet applications,” in Proc. ICWAMTIP, pp. 276-281, 2012.

[14] ns-3 LTE Module, [on-line] https://www.nsnam.org/docs/models/html/lte-design.html, accessed July 2019

[15] M. Mezzavilla, M. Miozzo, M. Rossi, N. Baldo, and M. Zorzi, “A lightweight and accurate link abstraction model for the simulation of LTE networks in ns-3,” in Proc. ACM Int. Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2012.

[16] J. Dormand and P. Prince, “A family of embedded Runge–Kutta formulae,” Journal of Computational and Applied Mathematics, vol. 6, pp. 19–26, 1980.

[17] N. Tian, Z. Zhang, Vacation Queueing Models: Theory and Applications, Springer Science & Business Media, 2006.