End-To-End KPI Analysis in Converged Fixed-Mobile Networks

M. Ruiz, M. Richart, A. Castro, and L. Velasco*
Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail: lvelasco@ac.upc.edu*

ABSTRACT

The independent operation of mobile and fixed network segments is one of the main barriers that prevents improving network performance while reducing capital expenditures coming from overprovisioning. In particular, a coordinated dynamic network operation of both network segments is essential to guarantee end-to-end Key Performance Indicators (KPI), on which new network services rely on. To achieve such dynamic operation, accurate estimation of end-to-end KPIs is needed to trigger network reconfiguration before performance degrades. In this paper, we present a methodology to achieve an accurate, scalable, and predictive estimation of end-to-end KPIs with sub-second granularity near real-time in converged fixed-mobile networks. Specifically, we extend our CURSA-SQ methodology for mobile network traffic analysis, to enable converged fixed-mobile network operation. CURSA-SQ combines simulation and machine learning fueled with real network monitoring data. Numerical results validate the accuracy, robustness, and usability of the proposed CURSA-SQ methodology for converged fixed-mobile network scenarios.
Keywords: converged fixed-mobile networks, real-time KPI estimation.

1. INTRODUCTION

Fixed-mobile networks have been traditionally operated as two separated network segments, where the Evolved Packet Core (EPC) in the Radio Access Network (RAN) facilitates the mobility of User Equipment (UE) and provides Quality of Service (QoS) looking at meeting the needs of services and users with diverse characteristics, whereas the fixed network provides connectivity services among Evolved NodeB (eNB) / Next Generation NodeB stations and with the mobile core. Although this separation simplifies network operation, it imposes resource overprovisioning to the fixed network; enough resources need to be allocated in the fixed network trying to avoid network congestion that would degrade the QoS perceived by the end-users (i.e., increased end-to-end delay and reduced throughput). Note that overprovisioning increases network capital expenditures (CAPEX).

Large traffic variations can be expected not only in the RAN but also in the fixed network as a result of the increment of the bitrate available in the RAN, the different type of services (e.g., video streaming, P2P, gaming, and so on), and the mobility of UEs. Such traffic variations push the amount of resources to be overprovisioned in the fixed network. In addition, the stringent requirements imposed by 5G is making that fixed networks need to be redesigned while fostering the convergence of mobile and fixed networks, where the optical transport network is extended toward the edge. In this regard, operators are attending with significant interest to the definition of the next-generation cell site Gateway (CSGw) connecting current and upcoming 5G mobile cell sites, to the transport network; the CSGw includes, among others, Multiprotocol Label Switching (MPLS) capabilities, so traffic engineering techniques can be applied to *packet flows*. Note that MPLS improves the routing and increases the traffic engineering possibilities and it can be used to implement the data plane, e.g., to support the S1 interface [1]. In this context, looking at limiting CAPEX derived from the required overprovisioning, the convergence of mobile and fixed networks needs to be complemented with some level of coordination at the control plane.

Given this, we propose a tool that uses traffic prediction and UEs' mobility as inputs to compute future network conditions and estimate QoS-related end-to-end KPIs for the current network configuration. In this respect, in our previous work in [2], we proposed a methodology named CURSA-SQ to analyze traffic flows in a fixed network by modelling service traffic and the behavior of the queues in packet nodes. CURSA-SQ enables near real-time traffic analysis (with sub-second granularity) due to its better performance and scalability compared to traditional discrete-event based simulations. Starting from the general CURSA-SQ methodology, in this paper, we present the needed extensions to enable its application in converged fixed-mobile network scenarios, where each cell in the RAN is modeled as a shared medium controlled by the cell's scheduler.

2. ESTIMATING END-TO-END KPIS IN A FIXED-MOBILE NETWORK

Figure *1*a illustrates the considered fixed-mobile network scenario, where eNBs in the RAN are connected to packet nodes in the fixed access-metro network through CSGws. In the control plane, we assume an SDN controller in charge of the access-metro network that includes CSGws and packet nodes, as well as an MDA controller collecting monitoring data from the access-metro nodes. To support the S1 interface between eNBs and the mobile core, MPLS tunnels can be set-up in the access-metro network to facilitate traffic flow management (see [1]).

Aiming at enabling near real-time access-metro and mobile KPI estimation, monitoring data collected continuously from network devices can be used for analysis. However, variations in the input traffic to the access-metro network due to users' activity and mobility requires data that is available in the EPC with a full view of the cells. In addition, the evaluation of mobile services requires analyzing the behavior of the access-metro network ML models can be fed with these data to forecast relevant variables, e.g., the number of active UEs, their position, and their mobility among cells within the next short time window (e.g., next 1-2 minutes). With such prediction, as well as with some known deterministic network parameters, CURSA-SQ can be used to simulate network conditions in a future time window and evaluate KPIs in every networking device, as well as end-to-end. Such evaluation, together with some recommendations, can be of paramount importance for the fixed and the mobile network operation.

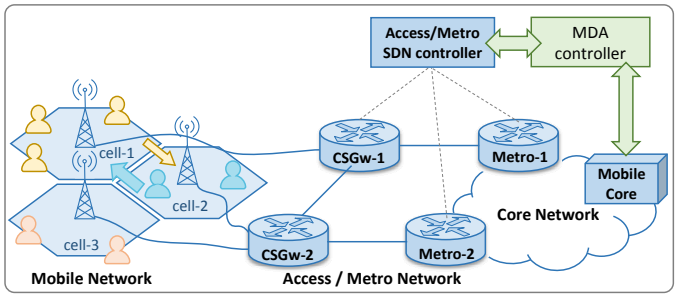

Figure 1. Converged fixed-mobile network.

Although measurements of the amount of bitrate entering every interface of a node in the access-metro network can be provided (in particular, those connecting the eNBs in the RAN), they are not enough to compute per-service and per-UE KPIs with enough accuracy. In fact, as such measurements would entail installing expensive deep packet inspection (DPI) devices to examine the contents of every packet entering the access-metro network, we assume that no DPI devices are installed. To overcome the lack of per-service and per-UE real-time measurements, CURSA-SQ includes a *dynamic configuration* module that, among other tasks, finds a feasible traffic disaggregation given the aggregated measured traffic and the information related to the UEs in the RAN; specifically, likely per-service flows are estimated, so that their summation produces an aggregated estimated flow that statistically behaves similar to the aggregated measured one. With such estimated traffic disaggregation, the dynamic configuration module prepares the scenario to run a simulation phase for the next short time window, and an *end-to-end KPI estimation* module computes the KPIs based on the results of the simulation phase that are sent to the *performance analysis* module in the MDA controller; the latter can carry out some evaluation on the estimated end-to-end KPIs and send timely recommendations to the SDN controller and mobile core. Last but not least, an *evaluation and tuning* module waits until real aggregated monitoring data measured from the network is available, compares them to the results estimated by the *simulation* module for the same time period, and uses the results to tune specific parameters in the dynamic configuration module.

3. COMPUTING KPIS ON A MOBILE NETWORK

In this section, we extend the CURSA-SQ queue model to include shared medium and mobility. The CURSA-SQ queue model [2] is a continuous G/G/1/k queue model with a First-In-First-Out (FIFO) discipline [3] based on the logistic function.

Let us consider a scenario with a single cell and several UEs connected; all the traffic between UEs and the base station shares the same physical medium, and thus, its capacity. It is clear that the capacity of the shared medium is not evenly distributed among the UEs in the cell, as the capacity that every UE perceives depends on the signal-to-interference-plus-noise ratio (SINR) and thus, on its specific geo-localization. In this regard and aiming to reduce the size of the problem and thus its computation time, UEs can be aggregated into groups following a *similarity* criterion in terms of e.g., their perceived capacity or their perceived signal quality. Note that the similarity criterion should be considered together with the cell's scheduler in order to achieve the most accurate results. In this paper, we assume the *proportionally fair* (PF) policy [4] and group UEs by the similarity in their perceived capacity. The size of the groups varies with time as UEs move, so mobility is modeled by updating the size of two or more groups.

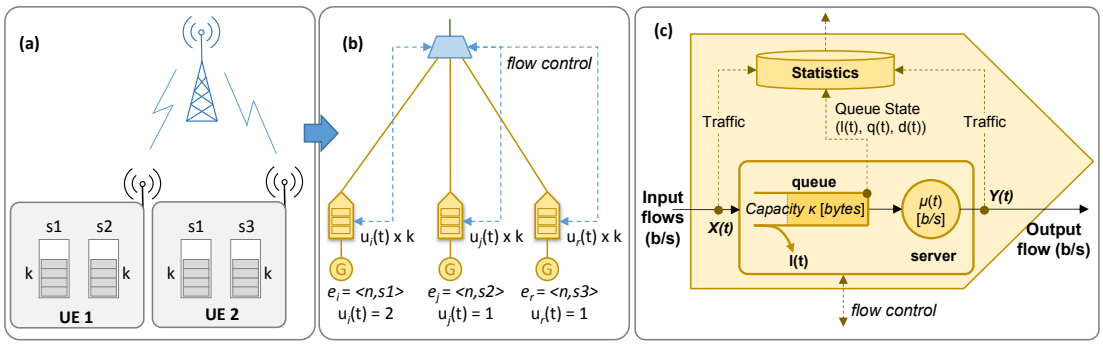

Figure 2. Example of cell modeling of two UEs in a group (a-b). CURSA-SQ queuing model for a mobile entity (c).

According to the general CURSA-SQ methodology [2], 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 modeled as a system of queues, where we define a different queue for each group of UEs consuming the same service (see Figure 2a-b). Then, given a set S of services and a set N_c of groups of UEs, a cell c is modeled as a set E_c of $|N_c|\times|S|$ mobile entities, each with a traffic generator and a queue. Let us now define the traffic generated by a mobile entity $e = <n, s>$, which can be characterized by the number of UEs $u(t)$ in group n and the traffic profile defining service s. In particular, two random variables are used to model the traffic of every single UE related to a given service s: *i*) the inter-arrival burst rate, and *ii*) the burst size. Then, the expectation and variance of both random variables are conveniently scaled using $u(t)$ to generate aggregated traffic traces (see [2] for more details).

As for mobility, as introduced above, it is modeled by updating the value of $u(t)$ of two or more entities in a correlated manner. Note that by updating $u(t)$, both intra- and inter- cell mobility can be implemented, depending on whether the entities are in the same or in different cells.

Regarding queues, they are characterized by their capacity and their server rate. The capacity of the queue in an entity e is given by the buffer size (k) typically allocated by the Radio Link Control (RLC) protocol [4] times the number of UEs $u(t)$ in the entity. As for the server rate μ, we redefine that in the original CURSA-SQ model as being a function of time ($\mu(t)$), which converts the system into a *non-autonomous* Ordinary Differential Equation (ODE). In the cell model in Figure 2b, all the queues are connected to an element that aggregates and disaggregates the traffic of the cell and emulates the shared medium of the cell, while implementing the PF scheduler; specifically the aggregator implements flow control by tuning the value of the server rate $\mu_e(t)$ for each mobile entity e in the cell c. Then, $\mu_e(t)$ can be modeled as:

$$\mu_e(t) = C_c \cdot [\alpha_c(t) \cdot g(q_e(t), q(t)) + (1 - \alpha_c(t)) \cdot f_e], \quad (1)$$

where: *i*) $\alpha_c(t)$ is a weighting factor in [0,1]; *ii*) $g(\cdot)$ models the cell's scheduler policy with $q_e(t)$ being the current state of the local queue and $q(t)$ that of all queues in the cell; and *iii*) f_e is the fixed proportion of the capacity of the cell that mobile entity e will perceive, computed as SINR_e / $\sum_{e' \in Ec}$ $\text{SINR}_{e'}$. The scheduler can manage the capacity sharing among the different mobile entities as a function of every request.

We implement the PF scheduler by solving the problem for every single cell for every time t in the given time window in two stages: *i*) the initial stage (*stage 0*) assumes $\alpha_c = 0$, i.e., $\mu_e(t)$ is proportional to the SINR perceived by entity e. This stage returns a value for $q_e(t)$, denoted $q^0{}_e(t)$; *ii*) the second stage uses a value of α_c that balances the server bitrate assigned to each entity in order to introduce fairness. α_c can be estimated as eq. (2):

$$\alpha_c(t) = \frac{1}{|N_c| \cdot |S| \cdot k} \cdot \sum_{e \in Ec} q_e^0(t). \quad (2)$$

As evaluating α_c for every time t in the given time window would heavily impact on the performance of the proposed method, we run stage 0 for longer periods where α_c is kept constant, and transform eq. (2) by computing the maximum for the period. Note that eqs. (1) and (2) focus on providing fairness among the entities in the cell; the resulting $\mu_e(t)$ value is evenly shared by all the UEs in the entity, and hence to preserve fairness among UEs, size of the entities should be balanced.

Figure 2c shows the details of the queue for mobile entities; it stores the incoming flow traffic in a queue of capacity k, where the flow remains until it leaves at the programmed server rate $\mu_e(t)$. By solving the CURSA-SQ model with the shared medium extension in eq. (1) for a time interval, *per-entity* throughput, delay, packet loss and others KPIs can be obtained; *per-UE KPIs* are computed proportionally from *per-entity* ones. Note that the server rate becomes $\mu_e(t)$ now.

4. RESULTS

In this section, we numerically study and validate the proposed CURSA-SQ methodology. To this aim, we focus on evaluating the performance of the proposed extension for shared medium by means of the ns-3 network simulator. To validate CURSA-SQ extension for shared medium, we run several simulations to compare the performance of our Matlab implementation against that of the ns-3 network simulator implementing the LTE module modeling the full LTE Radio Protocol and the EPC, including the core network interfaces, protocols, and entities; both run on an i7-Gen8 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

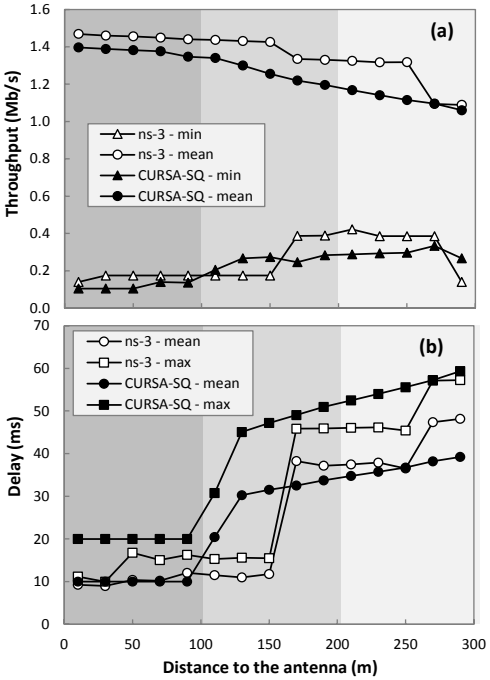

Figure 3. Performance of the radio segment.

packets) injected by a random bursty traffic generator. Each sector was modeled 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 [5], the simulator finds the best *modulation and coding scheme* for a given channel condition. For the sake of a fair comparative analysis, components f_e and $g(\cdot)$ in eq. (1) have been modeled to match the abovementioned configuration. In addition, packet traces generated during ns-3 simulation were aggregated in flows with a granularity of 250 ms and used in the CURSA-SQ simulation.

Figure *3* 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 Figure 3a and Figure 3b, 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 – 200 m), 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.

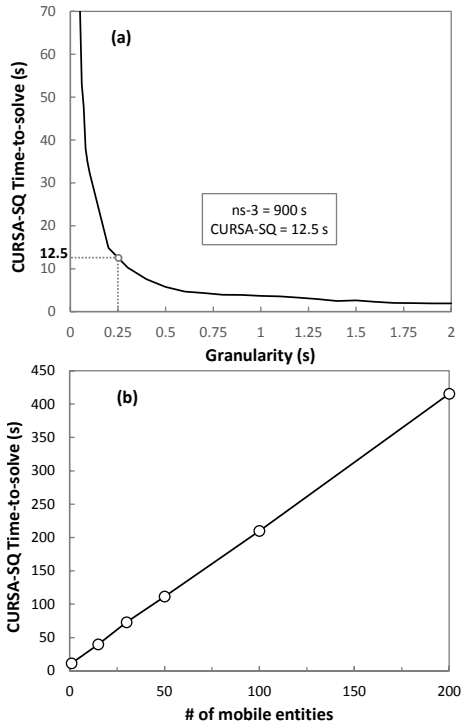

Figure 4. CURSA-SQ scalability.

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. Figure *4*a 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 250 ms 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. Assuming such granularity, Figure 4b shows the CURSA-SQ time-to-solve when the number of mobile entities in a cell increases; the results show a clear linear trend that is related to the number of calls to the ODE solver, which confirm the applicability of CURSA-SQ for a wide range of realistic scenarios.

5. CONCLUSION

There is a clear need to estimate end-to-end KPI's in scenarios of converged fixed-mobile networks as the key to verify the performance of services. In this context, an extension to the general CURSA-SQ methodology for shared medium and mobility features has been presented, where UEs are grouped into entities as a function of the SINR that they perceive and the service that users consume. Entities are modeled as queue systems, where the service rate varies with time and depends on the actual SINR and the cell's scheduler policy.

CURSA-SQ was validated against the ns-3 network simulator for a pure mobile network scenario. CURSA-SQ estimations probed to be accurate while simulating 120s with granularity 250ms in just 12.5s. These results highlight the outstanding scalability of the proposed method for near real-time computation.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Spanish MINECO TWINS project (TEC2017-90097-R), and from the Catalan Institution for Research and Advanced Studies (ICREA).

REFERENCES

[1] F. López, U. Silva, D. Campelo, R. Oliveira, S.-J. Lim, and L. García, "QoS management and flexible traffic detection architecture for 5G mobile networks," *Sensors*, vol. 19, pp. 1335, 2019.

[2] 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.

[3] U. Bhat, "*An Introduction to Queueing Theory: Modeling and Analysis in Applications*," Birkhäuser Basel, 2015.

[4] S. Sesia, I. Toufik, and M. Baker, "*LTE – The UMTS Long Term Evolution – From Theory to Practice*," Wiley, 2009.

[5] M. Mezzavilla *et al*., "A lightweight and accurate link abstraction model for the simulation of LTE networks in ns-3," in *Proc. ACM Int. Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2012.