# A data-driven scheduler performance model for QoE assessment in a LTE radio network planning tool

P. A. Sánchez[a,*], S. Luna-Ramírez[a], M. Toril[a], C. Gijón[a], J. L. Bejarano-Luque[a]

[a]*Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, 29010, Málaga, Spain.*

**Abstract**

The use of static system-level simulators is common practice for estimating the impact of re-planning actions in cellular networks. In this paper, a modification of a classical static Long Term Evolution (LTE) simulator is proposed to estimate the Quality of Experience (QoE) provided in each location on a per-service basis. The core of the simulator is the estimation of radio connection throughput on a location and service basis. For this purpose, a new analytical performance model for the packet scheduling process in a multi-service scenario is developed. Model parameters can easily be adjusted with information from radio connection traces available in the network management system. The simulation tool is validated with a large trace dataset taken from a live LTE network.

*Keywords:* QoE, LTE, SON, BigData, Traces
*2018 MSC:* 00-01, 99-00

## 1. Introduction

Over the years, different radio access technologies and architectures have been developed with the aim of providing larger cell capacity, higher peak data rates and lower latency. In parallel, the increase of network and service complexity has turned network management into a very complex task. To alleviate this situation, Self-Organizing Networks (SON) [1] aim to automate labor-intensive tasks in network planning and optimization procedures.

Legacy SON solutions adopted a network-centric approach focused on network performance. However, the latest technological advances in multimedia services have forced operators to move to a user-centric approach focused on Quality of Experience (QoE) [2]. In this context, the validation of any new SON algorithm has become extremely complex, as QoE is the result of many interrelated factors. The most straight-forward approach is to test the algorithm in field trials, but these are only carried out in small geographical areas and with very restrictive conditions for safety reasons. In the absence of the real network, network simulators allow different tests to check algorithm performance before implementation in the live network.

Mobile simulation tools are divided into link-level and system-level simulators. Link-level simulators (e.g., [3, 4, 5]) are focused on the physical layer, and thus often only model one transmitter-receiver pair. In contrast, system (a.k.a. network) simulators (e.g., [6, 7, 8, 9]) provide a global system overview by including multiple users/cells, so that relevant network performance indicators (e.g., average cell throughput or call dropping ratio) can be obtained by network operators. For this purpose, models for higher protocol layers are included. For simplicity, only a limited set of network features are considered, together with simple performance models of the link layer.

Network simulators can also be static or dynamic. In the static approach, network performance is evaluated in specific instants without any time correlation between them. This can be done iteratively by generating different network states with connections randomly distributed by a Monte-Carlo method [10], or in one go by estimating the performance per location in a grid-based scenario [11]. In contrast, in the dynamic approach, system performance is evaluated by checking network evolution over time through a series of states depending on previous states. Dynamic simulators are therefore used to check the capability of radio resource management and self-tuning algorithms to react to changing system conditions [12, 13, 14, 15], whereas static simulators are preferred for self-planning due to their lower computational load [16, 17, 18, 19, 20] .

For computational efficiency, most current radio network planning tools include a static system-level simulator that follows a grid-based approach, where link performance at every single location is calculated at the same time. Thus, it is very difficult to model the radio resource assignment procedure (i.e., packet scheduling), so that link performance is only estimated from indicators of lower layers (e.g., Signal to Interference plus Noise Ratio, SINR [21]). Such an approach can only be used to derive upper bounds of link capacity with the Shannon formula [22]. In the absence of a better approach, it is generally assumed that the whole system bandwidth is assigned to the user (i.e., a single user in the scheduler). This is rarely the case in live networks, where multiple users simultaneously demand re-

*Corresponding author
*Email addresses:* pso@ic.uma.es (P. A. Sánchez), sluna@ic.uma.es ( S. Luna-Ramírez), mtoril@ic.uma.es ( M. Toril), cgm@ic.uma.es ( C. Gijón), jlbl@ic.uma.es (J. L. Bejarano-Luque)

sources. Even if analytical performance models can be derived for simple multi-user schedulers (e.g., Round Robin, Proportional Fair...), these models cannot easily be extended to the multi-service case, where radio schedulers assign different radio resources not only depending on radio link performance, but also on user traffic demand at every time transmission interval. As a result, most network planning tools fail to give realistic user throughput or QoE figures, nor include service differentiation.

Data from network management processes can be used to calibrate traffic and propagation models with live performance statistics. The simplest approach is to use performance counters gathered by base stations to tune model parameters [23]. In LTE systems, relevant counters are broken down by cell and service class (Quality of Service Class Identifier, QCI) [24]. With recent advances in big data technologies, it is now possible to collect and analyze very detailed information from signaling events in the network, referred to as traces. These reflect the performance of individual connections, and can thus be used to adjust simulation tools [25]. In [26], the authors propose a method to improve the accuracy of radio network utilization measurements in a live LTE network based on connection traces. The result is an accurate map with the spatio-temporal distribution of downlink resources for a particular scenario. However, no performance model is provided to update the distribution after changes in the environment (e.g., antenna tilt change, new traffic hot spot...). To the authors' knowledge, no previous work has considered the use of traces to tune QoE-related performance models in a mobile network planning tool.

In this work, a new performance model of the scheduling process is proposed for a static grid-based LTE network simulator. The core of the model is the estimation of the average amount of resources assigned to a user of a service in each location, from which to derive user radio throughput on a per-location and service basis. Unlike previous analytical works, the proposed model is adjusted with connection traces from a live network. The main contributions of this work are: a) a data-driven scheduler performance model that can be used to build QoE maps for a particular scenario with a radio network planning tool, and b) a comprehensive performance analysis based on a real trace dataset.

The structure of the paper is as follows. Section 2 outlines the processing of traces for the simulation tool. Section 3 describes the scheduler performance model used to obtain user throughput and QoE estimates. Section 4 presents model assessment in a realistic scenario. Finally, Section 5 summarizes the main conclusions of the work.

## 2. Traces

Mobile networks generate a large amount of information that can be used in measurement-based re-planning and optimization tasks [27]. In the radio access domain, such information can be classified into:

1. Configuration Management (CM) information, consisting of network parameter settings.
2. Performance Management (PM) information, consisting of counters that collect aggregated measurements reflecting the performance of network elements. These counters are used to compute network Key Performance Indicators (KPI).
3. Traffic Recordings (TR), also known as *traces*, collecting signaling messages exchanged between the different network nodes. These are divided into:
   (a) Cell Traffic Recordings (CTR), containing, anonymously, events and measurements of a pre-established percentage of connections in a cell.
   (b) User Equipment Traffic Recordings (UETR), with events and measurements for a specific user.

Trace files are binary files containing signaling events. The structure of events consists of a header with general attributes (e.g., timestamp, network node, user equipment, event type or event length) and a message container including different attributes (a.k.a. event parameters). Events include vendor-dependent internal events, generated inside the base stations for monitoring purposes, and standardized external events, corresponding to messages exchanged between network nodes. Event decoding is performed by a parsing tool that extracts the information contained on fields per event type, network node and reporting period. Then, traces are synchronized by merging files from different nodes. The reader is referred to [26, 28] for more details on trace processing.

One of the main limitations of the current set of KPIs is the lack of indicators segregated by application. As explained above, counters are only broken down by cell and service class (QCI). Likewise, trace records can be segregated by QCI. Unfortunately, some service classes comprise applications of very different nature. For instance, QCI8 in live networks may include Transmission Control Protocol (TCP)-based Video streaming, File Transfer Protocol (FTP) and chat services. Thus, a more elaborated segregation approach is needed.

Several works have addressed traffic classification based on connection descriptors. This can be done by analyzing the first packets of the connections (early classification [29]) or the whole connection (late classification [30, 31]). Clustering can be done by machine learning algorithms, which can be divided into supervised, semi-supervised and unsupervised learning algorithms. In this work, an unsupervised clustering algorithm is used for clustering connections offline based on its attributes in the radio interface, in the absence of a trace dataset with labeled cases that can be used to train the classification model. In particular, the k-medoids algorithm is selected, as it is more robust against outliers than the classical k-means algorithm [32] . The process is as follows:

1. First, events in traces are decoded and synchronized. As a result, a single file is generated with all events of the same type from all network nodes (cells). In this file, each event corresponds to a different user and cell. Then, events are segregated by event type into separate files.
2. From the synchronized events, a connection is defined for each user connected to a cell by isolating the start/end events of the Radio Resource Control (RRC) connection.

Table 1: Connection indicators for traffic segregation

| Name | Unit |
|---|---|
| RRC connection time, $T_{con}$ | ms |
| DL data volume, $V^{DL}$ | bytes |
| UL data volume, $V^{UL}$ | bytes |
| No. of active TTIs in DL, $N_{act}^{DL}$ | ms |
| No. of active TTIs in UL, $N_{act}^{UL}$ | ms |
| Last TTI data volume in DL, $V_l^{DL}$ | bytes |

Then, events with performance measurements associated to the connection are included as attributes in the radio interface.

3. To ensure that all indicators are equally important when calculating medoids, indicators are normalized. Then, group analysis is performed. K-medoids starts by selecting $K$ points as initial medoids. Points are assigned to the group represented by the nearest medoid. Then, a non-medoid point and its closest medoid are swapped, points are reassigned and the overall cost of selecting that point as a medoid is calculated. The cost is measured using the Mean Average Error (MAE) metric, calculated as

$$MAE = \sum_{k=1}^{K} \sum_{x_i \in c_k} |c_k - x_i| \,, \tag{1}$$

where $k$ represents the group index, $K$ is the number of groups (selected a priori), $c_k$ the position of the medioid of group $k$ and $x_i$ the position of the current point, belonging to group $k$. If the cost of selecting the new point as medoid of its group is lower, the change is maintained. The process is repeated until medoids do not change or a previously defined convergence criterion is met (e.g., maximum number of iterations).

Table 1 shows indicators selected a priori as input for traffic classification. Connection duration is defined by the RRC connection time. Data volume is included for both the complete connection and the last Transmission Time Intervals (last TTIs). Last TTI refers to those TTIs when the transmission buffer of the User Equipment is emptied. In these TTIs, the user does not take advantage of the whole capacity that could potentially be assigned by the scheduler. Such piece of information can be used as a rough indicator of traffic burstiness. To better differentiate services, data volume figures are broken down in uplink (UL) or downlink (DL).

As shown later, only 3 service groups are defined a priori to reduce the number of applications handled by the simulator: Social network/Web browsing, Application download and Video streaming. Voice over LTE (VoLTE) service is not included here, due to the absence of VoLTE traffic in the real dataset used in this work. This does not imply a loss of generality, since, unlike the rest of services, VoLTE can easily be segregated, since it is often the only service with QCI $= 1$ in current networks [24].

Table 2: Notations of the equations of the packet scheduling performance model

| Parameter | Definition | Unit |
|---|---|---|
| $BW(u)$ | Average bandwidth assigned to user u | Hz |
| $BW(u)_{PRB}$ | Bandwidth of a PRB (15 kHz) | Hz |
| $\overline{N_{PRB}}(u)$ | Average number of PRBs per TTI assigned to the user | PRB |
| $\overline{N_{PRB}}(s,u)$ | Average number of PRBs per TTI assigned to user $u$ with service $s$ | PRB |
| $\overline{N_{PRB,n}}(u)$ | Average number of PRBs assigned to user $u$ as normal TTI | PRB |
| $\overline{N_{PRB,l}}(s,u)$ | Average number of PRBs assigned to user $u$ and service $s$ as last TTI | PRB |
| $N_{TTI,ROP}(u,s)$ | Number of active TTIs in the ROP for user $u$ with service $s$ | TTI |
| $N_{PRB}(c)$ | Total number of available PRBs in cell $c$ | PRB |
| $N_u(c)$ | Number of simultaneous TTI users in cell $c$ | users |
| $N_{u,last}(c)$ | Number of simultaneous last TTI users in cell $c$ | users |
| $N_{u,norm}(c)$ | Number of simultaneous normal TTI users in cell $c$ | users |
| $N_{con}(c,s)$ | Number of connections with service $s$ in cell $c$ | conn |
| $N_{con,TOT}(c)$ | Number of connections in cell $c$ | conn |
| $R_n(c,s)$ | Average time ratio of normal TTI for service $s$ in cell $c$ | - |
| $R_l(c)$ | Average time ratio of last TTI in cell $c$ | - |
| $R_l(c,s)$ | Average time ratio of last TTI for service $s$ in cell $c$ | - |
| $SE(u)$ | Spectral efficiency of the user | $\frac{\text{bps}}{\text{Hz}}$ |
| $SINR(u)$ | Average Signal to Noise plus Interference Ratio for the user | - |
| $TH(u)$ | Average throughput experienced by user $u$ | bps |
| $TH(u,s)$ | Average throughput experienced by user $u$ with service $s$ | bps |
| $V_{ROP}(u,s)$ | DL data volume during ROP for user $u$ with service $s$ | Bytes |

## 3. System model

In this section, a packet scheduling performance model for a grid-based static LTE network simulator is described. As in other grid-based simulators, performance estimates correspond to a snapshot reflecting the average system state. Thus, no user mobility is considered. Since the aim is to provide QoE indicators, QoE models for the different services are also presented. The model is valid for both UL and DL, although only DL performance is evaluated here, since most traffic in current networks is carried in that link.

### *3.1. Packet scheduling performance model*

For a clearer reading, Table 2 summarizes all parameters used for the definition of this performance model. User transmission rate can be approximated by Shannon's formula, as

$$TH(u)\,[\text{bps}] = BW(u)\,[\text{Hz}] \cdot SE(u)\left[\frac{\text{bps}}{\text{Hz}}\right] \approx N_{PRB}(u)\,[\text{PRB}] \cdot BW_{PRB}\left[\frac{\text{Hz}}{\text{PRB}}\right] \cdot \log_2(1 + SINR(u)[-])\,, \tag{2}$$

where $TH(u)$ is the average throughput experienced by user $u$, $BW(u)$ is the average bandwidth assigned to the user, $SE(u)$ is the spectral efficiency of user $u$, $N_{PRB}(u)$ is the average number of PRBs per TTI assigned to user $u$, $BW_{PRB}$ is the bandwidth of a Physical Resource Block (PRB) (i.e., $BW_{PRB}$=15 kHz ) and $SINR(u)$ is the average signal-to-interference-plus-noise ratio for user $u$. Note that, in a grid-based static simulator, every location in the scenario is considered as a potential user, so that network performance has to be calculated in all possible locations in the scenario. Likewise, it is assumed that all users in the same location obtain the same network performance (provided that they demand the same service). Thus, $TH(u)$, $BW(u)$, $N_{PRB}(u)$ and $SINR(u)$ represent values at a given location in the scenario, and $u$ denotes a specific user position. Once user performance is evaluated at every position,

cell and network performance is calculated by a weighted average across locations. Finally, it is assumed that every position/user is served by a single cell $c$ providing the highest pilot signal level in that location. Due to the difficulty of modeling the resource allocation process (i.e., $N_{PRB}(u)$), most radio planning tools assume that a user always receives the full system bandwidth (i.e., a single user with data to transmit in the scheduler). Such an optimistic assumption makes that network KPIs should only be considered as an upper performance bound only valid for lightly loaded cell. To solve this limitation, in this work, $N_{PRB}(u)$ is estimated for each type of service and position (i.e., $N_{PRB}(u, s)$) with the real load conditions.

The amount of resources assigned to a user strongly depends on the user data volume in the transmission buffer. Such an influence must be taken into account when evaluating the performance of a packet scheduler. Otherwise, a low throughput due to lack of data may be wrongly interpreted as a bad link performance. This effect is critical in those services whose connections last a few TTIs (e.g., social networks, chat applications...). To account for this dependency, the analysis is broken down into normal TTI and last TTI users. A normal TTI user is one that still has data in the buffer after transmitting in the TTI under consideration. In contrast, a last TTI user is one having the buffer momentarily empty after transmitting in the TTI under analysis (either because the user is waiting for new data bursts or the connection has ended). Figure 1 illustrates the concept of normal and last TTIs in a connection. Three connections (i.e. users) are included in the figure. Upper part in the figure represents data in user transmission buffer for every connection. Every TTI, the scheduler (lower part in the figure) assigns a number of PRBs to some/all users, so data in buffer for those users decrease. This is the case, for example, of connections 1 and 3 in TTI $n$ (while connection 2 has no PRBs assigned in TTI $n$). Additionally, user data may increase in a TTI due to a new data burst from user traffic source. This is the case of connection 1 in TTI $n+3$ and connection 3 in TTI $n+2$. Thus, a user is experiencing a normal TTI when its data buffer is not empty for that TTI (e.g., TTI $n$ and $n+3$ for connection 1 or TTI $n$, $n+1$ and $n+2$ for connection 2). A user is, however, experiencing a last TTI when its data buffer is emptied during that TTI (TTI $n+1$, $n+3$ and $n$ for connections 1, 2 and 3, respectively, in the figure 1 )

For simplicity, it is assumed that: a) last TTI users have preference over normal TTI users in the scheduler, receiving all resources they need (i.e., only compete with other last TTI users), and b) the resources left by last TTI users are equally shared by normal TTI users, as in a Round Robin (RR) allocation scheme. In the time domain, a user would be considered as a normal TTI user for several consecutive TTIs and as a last TTI user for one (i.e., the last) TTI in every data burst. These alternating states can only be considered in the form of time ratios when evaluating a snapshot in a static grid-based simulator. Thus, the average number of PRBs assigned per TTI to a user $u$ for a service $s$, $\overline{N_{PRB}}(s, u)$, is estimated as the average number of PRBs received as normal and last TTI user, calculated as

$$\overline{N_{PRB}}(u, s) = R_n(c, s)\overline{N_{PRB,n}}(u) + R_l(c, s)\overline{N_{PRB,l}}(u, s)\ , \quad (3)$$

where $c$ is the serving cell in the location of user $u$ for service $s$, $\overline{N_{PRB,n}}(u)$ and $\overline{N_{PRB,l}}(s, u)$ are the average number of PRBs assigned to user $u$ as normal and last TTI user for service $s$, respectively, and $R_n(c, s)$ and $R_l(c, s)$ are the average time ratios of normal/last TTIs for service $s$ in cell $c$. An inspection of (3) shows that the average number of PRBs assigned to the user depends on the amount of resources assigned to the user in each state and the normal/last TTI ratio . Note that $\overline{N_{PRB,l}}(s, u)$ depends on user position and demanded service $s$, whereas $\overline{N_{PRB,n}}(u)$ only depends on user position, since it is assumed that resources in normal TTIS are assigned based on a RR scheme. Also important, $R_n(c, s)$ and $R_l(c, s)$ can be directly obtained from connection traces. The rest of this section details the calculation of $\overline{N_{PRB,n}}(u)$ and $\overline{N_{PRB,l}}(s, u)$.

### 3.1.1. Normal TTI model

For tractability, it is assumed that normal TTI users are assigned resources by a RR scheme without service prioritization. Thus, the amount of resources assigned to a normal TTI user served by a cell $c$ only depends on the number of PRBs available for normal TTI users and the number of simultaneous normal TTI users in cell $c$. Thus, $\overline{N_{PRB,n}}(u)$ is calculated as

$$\overline{N_{PRB,n}}(u) = \mathbb{E}\left[\frac{N_{PRB,cell}(c) - \sum_{u \in u_l(c)} N_{PRB}(u)}{N_u(c) - N_{u,l}(c)}\right]\ , \quad (4)$$

where $N_{PRB,cell}(c)$ is the total number of available PRBs in cell $c$, given by the system bandwidth ($N_{PRB,cell}(c) = 50\ \forall\ c$ in this work), $\sum_{u \in u_l(c)} N_{PRB}(u)$ is the number of PRBs assigned to last TTI users in cell $c$, $N_u(c)$ is the number of simultaneous active users (i.e., with data to transmit, including both normal and last TTI users) in cell $c$ and $N_{u,l}(c)$ is the number of simultaneous last TTI users in cell $c$. An inspection of (4) shows that the numerator reflects the number of PRBs available for normal TTI users (i.e., those left by last TTI users) and the denominator reflects the number of simultaneous normal TTI users, $N_{u,n}(c)$. The mean operator in (4) operates in the time domain (i.e., across TTIs). By taking advantage of the linearity

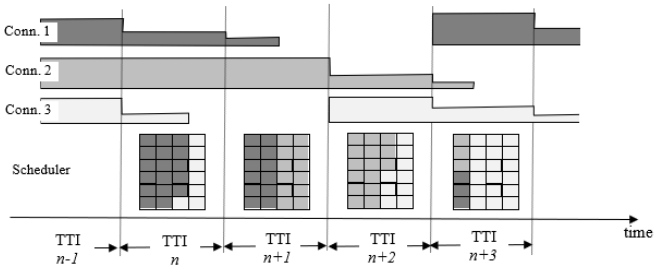



Figure 1: Three normal/last TTI connections in a scheduler.

Table 3: Calculation of probabilities for combinations of normal and last TTI users.

| $N_u(c)$ | $N_{u,n}(c)$ | $N_{u,l}(c)$ | $P$ |
|---|---|---|---|
| 1 | 1 | 0 | $(1-R_l(c,s))P\{N_u(c,s)=1\}$ |
| 2 | 1 | 1 | $(1-R_l(c))R_l(c)P\{N_u(c)=2\}$ |
| | 2 | 0 | $(1-R_l(c))^2P\{N_u(c)=2\}$ |
| 3 | 1 | 2 | $(1-R_l(c))R_l^2(c)P\{N_u(c)=3\}$ |
| | 2 | 1 | $(1-R_l(c))^2R_l(c)P\{N_u(c)=3\}$ |
| | 3 | 0 | $(1-R_l(c))^3P\{N_u(c)=3\}$ |

of the mean operator and the statistical independence between the numerator and denominator, (4) can be reformulated as

$$\overline{N_{PRB,n}}(u) = (N_{PRB}(c) - \overline{N_{PRB,l}}(c)) \cdot \mathbb{E}\left[\frac{1}{N_u(c) - N_{u,l}(c)}\right], \quad (5)$$

where $\overline{N_{PRB,l}}(c)$ is the average total number of PRBs assigned to last TTI users in cell $c$, which can be obtained from traces. Now, the mean operator is only applied to the inverse of the number of normal TTI users across TTIs.

In the absence of a closed-form expression, the inverse average in (5) is computed numerically. For this purpose, two constraints are introduced : a) $N_u(c) \geq 1$, since only TTIs with active users must be taken into account, and b) $N_{u,n}(c) \geq 1$, since the mean operator must be calculated only in those TTIs with normal TTI users. Table 3 shows all possible combinations of normal and last TTI users (considering only cases that satisfy a) and b)) in a cell $c$ for $N_u(c) = 1$, 2 and 3, together with their probability of occurrence, $P\{N_{u,n}(c), N_{u,l}(c))|N_{u,n}(c) \geq 1\}$, generalized as

$$P\{N_{u,n}(c), N_{u,l}(c)|N_{u,n}(c) \geq 1\} = \sum_{n_u=1}^{\infty} P\{N_u(c) = n_u\} \cdot \sum_{n_{u,n}=1}^{n_u} (1-R_l(c))^{n_{u,n}} R_l(c)^{n_u - n_{u,n}}, \quad (6)$$

where $P\{N_u(c) = n_u\}$ is the probability of $N_u(c)$ being $n_u$, $n_u \in \mathbb{N}$. $R_l(c)$ in (6) and Table 3 denotes the average ratio of last TTI users in the cell under consideration, including all services, i.e.,

$$R_l(c) = \sum_s R_l(c,s) \cdot \frac{N_{con}(c,s)}{N_{con,T}(c)}. \quad (7)$$

where $N_{con}(c,s)$ is the number of connections of service $s$ in cell $c$ and $N_{con,T}(c)$ is the total number of connections aggregating all services in cell $c$.

By extending the calculation to $N_u(c) \to \infty$, the inverse average in (5) is computed numerically as

$$\mathbb{E}\left[\frac{1}{N_u(c) - N_{u,l}(c)}\right] = \sum_{n_u=1}^{\infty} P\{N_u(c) = n_u | n_u \geq 1, n_{u,n} \geq 1\} \cdot \sum_{n_{u,n}=1}^{n_u} \frac{1}{(1-R_l(c))^{n_{u,n}}} \cdot \frac{1}{n_{u,n}} \cdot (1-R_l(c))^{n_{u,n}} \cdot (R_l(c))^{n_u - n_{u,n}}, \quad (8)$$

where $P\{N_u(c) = n_u\}$ is defined in (6).

The probability $P\{N_u(c) = n_u\}$ is the only parameter that cannot be obtained from traces. In this work, it is assumed that the arrival connection rate follows a Poisson distribution, so that

$$P\{n_u|n_u \geq 1\} = \frac{P\{n_u\}}{1 - P\{n_u = 0\}} = \frac{e^{-\overline{N_u}(c)}(\overline{N_u}(c))^{n_u}}{n_u!} \cdot \frac{1}{1 - e^{-\overline{N_u}(c)}}, \quad (9)$$

where $\overline{N_u}(c)$ is the average number of simultaneous users with data to transmit in the scheduler of the cell. Finally, recall that the RR algorithm distributes the PRBs available for normal TTI users in cell $c$ among those users, and, thus, the final PRB assignment does not depend on the specific location/user or service in a cell $c$, i.e., $\overline{N_{PRB,n}}(u)$ values only differs between users of different cells.

### *3.1.2. Last TTI model*

The calculation of $\overline{N_{PRB,l}}(u,s)$ implies a process similar to that of $\overline{N_{PRB,n}}(u)$, with a mean operation across time. Specifically, the average number of PRBs assigned to a last TTI user $u$ demanding service $s$ in cell $c$ can be expressed as

$$\overline{N_{PRB,l}}(u,s) = \mathbb{E}\left[\frac{TH_l(u,s)}{SE(u)}\right], \quad (10)$$

where $TH_l(u,s)$ is the throughput of user $u$ of service $s$ when transmitting in last TTIs and $SE(u)$ is the spectral efficiency of user $u$. Statistical independence between numerator and denominator allows to rewrite (10) as

$$\overline{N_{PRB,l}}(u,s) = \mathbb{E}\left[TH_l(u,s)\right] \cdot \mathbb{E}\left[\frac{1}{SE(u)}\right]. \quad (11)$$

The first factor (i.e., the average throughput experimented by a last TTI user) can be approximated as:

$$\mathbb{E}\left[TH_l(u,s)\right] \approx \frac{V_{ROP,l}(u,s)}{N_{TTI,ROP,l}(u,s)}, \quad (12)$$

where $V_{ROP,l}(u,s)$ is the DL data volume transmitted in last TTIs during the entire Reporting Output Period (ROP) for user $u$ of service $s$ and $N_{TTI,ROP,l}(u,s)$ is the number of last TTIs in the ROP for such user. Both indicators can be calculated from data traces.

For simplicity, the second factor in (11) (i.e., the average of the inverse of user spectral efficiency) is approximated as

$$\mathbb{E}\left[\frac{1}{SE(u)}\right] \approx \frac{1}{\mathbb{E}\left[SE(u)\right]} = \frac{1}{\overline{SE}(u)}\,. \tag{13}$$

### 3.2. QoE models

Once user throughput is estimated on a per-location and service basis, user QoE can be estimated. In this work, throughput figures are translated into QoE figures in a Mean Opinion Score (MOS) scale by means of the utility functions described in [33] for Social network/web browsing and App download and [34] for Video streaming.

For a Video streaming user, MOS is calculated as

$$MOS_{Video}(u) = 1 + (sQuality)\cdot \frac{\beta_1(sInteraction-1)+\beta_2(sView-1)}{4(\beta_1+\beta_2)}\,, \tag{14}$$

where $sQuality$ is the maximum MOS due to the Video quality, obtained from the tabulated values in [35] for a 5.5 inch screen, $sInteraction$ is the maximum MOS due to the initial loading time (in $[s^{-1}]$), $sView$ is the maximum MOS due to stall frequency and duration and $\beta_1$ and $\beta_2$ are regression constants with values $\beta_1 = 0.71$ and $\beta_2 = 0.77$ [35] . $sInteraction$ and $sView$ are obtained from [35] as a function of $L_{ti}$ and $L_{sr}$ respectively, where $L_{ti}$ is the average initial buffering time (in seconds) and $L_{sr}$ is the rebuffering ratio. In the radio planning tool, these indicators are obtained from radio throughput as [33]:

$$L_{ti}(u) = 5.91\frac{VBR(u)}{TH(u)} + 1.43\,, \tag{15}$$

$$L_{sr}(u) = \max(0, -0.915\frac{TH(u)}{VBR(u)} + 0.9667)\,, \tag{16}$$

where $VBR(u)$ is the average video bit rate of the sequence visualized by user $u$. The latter indicator is hard to estimate with current dynamic adaptive streaming schemes since it is not explicitly reported in connection traces. In the absence of a more precise approach, $VBR$ is estimated from $TH(u)$ as described in Table 4. Such an approach is in line with current adaptive streaming schemes that modify video resolution depending on buffer state. Note that the ratio $TH(u)/VBR(u)$ in (15)-(16) reflects how close/far is the actual user throughput from the target throughput required by the video source.

App download is modeled as a classical FTP service, whose MOS is estimated as

$$MOS_{FTP}(u) = \max(1, \min(5, 6.5\cdot TH(u) - 0.54))\,, \tag{17}$$

Table 4: Video bitrate estimation for Video streaming QoE model

| **TH(u)** [kbps] | **VBR(u) [kbps]** |
|---|---|
| $70000 \leq TH(u)$ | 10000 |
| $11500 \leq TH(u) < 70000$ | 4500 |
| $2800 \leq TH(u) < 11500$ | 2200 |
| $1200 \leq TH(u) < 2800$ | 1100 |
| $TH(u) < 1200$ | 700 |

where $TH(u)$ is user throughput (in Mbps).

Social network/web browsing is modeled as a classical web browsing service, whose MOS is estimated as

$$MOS_{web}(u) = 5 - \frac{578}{1+\left(\frac{TH(u)+541.1}{45.98}\right)^2}. \tag{18}$$

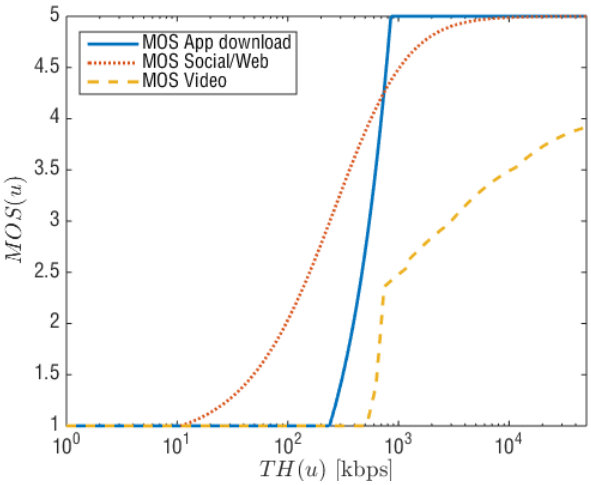



Figure 2: Utility functions.

For comparison purposes, Figure 2 plots the utility functions relating user throughput and MOS in each service. As expected, the same user throughput does not lead to the same QoE in all services.

## 4. Performance assessment

In this section, the proposed scheduler performance model is evaluated on a trace dataset taken from a live LTE network. For this purpose, the model is integrated into the grid-based static simulator described in [23]. For clarity, the simulation set-up is described first and results are presented later.

### 4.1. Analysis set-up

Assessment is carried out in a real scenario consisting of 129 macro LTE cells covering a downtown area of 150 km$^2$ in a large coastal city. Table 5 summarizes the main parameter settings in the static system-level simulator. For brevity, only DL performance is considered.

Table 5: Simulator Parameters

| Simulator Parameters | |
|---|---|
| Propagation Model | Okumura-Hata COST-231 |
| Grid Resolution [m] | 40 |
| Minimum propagation loss [dB] | 80 |
| Slow fading (std. deviation, $\sigma$) [dB] | 8 |
| DL carrier frequency [MHz] | 734 |
| UL carrier frequency [MHz] | 704 |
| System bandwidth [MHz] | 10 |
| Number of PRBs | 50 |
| **Base Station model** | |
| Maximun EIRP [dBm] | 46 |
| Number of sites | 44 |
| Number of cells | 129 |
| Spatial traffic distribution | Based on cell-level connection statistics and Timing Advance measurements |
| PRB utilization ratio [%] | [5, 70] |
| Avg. PRB utilization ratio [%] | 24 |
| **User Equipment model** | |
| Height [m] | 1.5 |

Table 6: Service performance statistics derived from traces

| Name | Social/Web | App download | Video | Total |
|---|---|---|---|---|
| $N_{con}(s)$ | 432128 | 75593 | 94406 | 602127 |
| $V_T^{DL}(s)$ [GB] | 5.96 | 472.04 | 49.53 | 528.28 |
| $V_{T,l}^{DL}(s)$ [GB] | 2.76 | 72.61 | 15.69 | 91.08 |
| $\overline{\left[\frac{V_l^{DL}}{V^{DL}}\right]}$ | 0.498 | 0.165 | 0.340 | 0.336 |
| $\overline{\overline{R_l}}(s)$ | 0.995 | 0.742 | 0.935 | 0.892 |
| $\max_c(\overline{R_l}(c,s))$ | 0.999 | 0.965 | 0.966 | 0.999 |

In the live network, trace collection is carried out for 1 hour in a working day, resulting in a dataset of 602127 connections. From this data, the parameters defined in Table 1 in the proposed scheduler performance model are obtained (or easily derived) on a cell and service basis. To have a global perspective of the scenario, Table 6 shows the values of those indicators broken down per service and aggregated across cells in the scenario for the busy hour of a working day. Specifically, the following indicators are included:

1. Total number of connections in the scenario per service, $N_{con}(s) = \sum_c N_{con}(c,s)$.
2. Total volume carried in DL in the scenario per service, $V_T^{DL}(s) = \sum_c V^{DL}(c,s)$.
3. Total volume transmitted in last TTIs in DL in the scenario per service, $V_{T,l}^{DL}(s) = \sum_c V_l^{DL}(c,s)$.
4. Average ratio of volume transmitted in last TTIs in DL across cells, $\overline{\left[\frac{V_l^{DL}}{V^{DL}}\right]} = \underset{c}{\text{avg}}\left(\frac{V_l^{DL}}{V^{DL}}(c,s)\right)$.
5. Average ratio of last TTIs across cells, $\overline{\overline{R_l}}(s) = \underset{c}{\text{avg}}(\overline{R_l}(c,s))$.
6. Maximum ratio of last TTIs across cells, $\max_c(\overline{R_l}(c,s))$.

First, it is observed that most connections in the scenario are tagged by the traffic classification algorithm as Video and Social network/Web services (16% and 72% of total connections, respectively) based on their features. Likewise, the average last TTI ratio (i.e., last TTIs/total active TTIs, $\overline{\overline{R_l}}(s)$) is extremely large (i.e., $\geq 90$ %) for all services but App download (74%). A closer analysis (not shown here) reveals that most connections consist of a single TTI. This large last TTI ratio supports the need for a model that considers last TTIs scheduling.

Model assessment is done by comparing three scheduler performance models: 1) a classical model, where all data is assumed to be transmitted in normal TTIs (i.e., $R_l(c)$=0 $\forall$ $c$) and users are assigned the full system bandwidth to transmit, hereafter referred to as Reference Scheduler Model (RSM), 2) an intermediate model, where PRBs in a cell are shared by $N_u(c)$ users with a RR resource allocation scheme, hereafter referred to as RR Scheduler Model (RRSM), and 3) the proposed model, described in Section 3, considering last TTI transmissions and service differentiation, hereafter referred to as Service Scheduler Model (SSM).

Models are compared in 2 different network load scenarios to check the potential of the proposed approach in different load conditions. First, models are compared with the real traffic and load distribution observed in traces to quantify the impact of considering last TTIs in a live scenario. This scenario is referred to as Measured Load (ML). Then, models are compared in the same scenario, but with a Low network Load (LL), generated by artificially reducing the number of connections in the network, setting the load of all the cells at 5%, with $\overline{N_u}(c) \approx 1$. The aim of including the LL scenario is to illustrate the need for considering last TTI transmissions specially in low load conditions, when almost all connections consist of last TTI transmissions due to a high radio resource availability.

In the LL scenario , the average number of simultaneous users per TTI per cell, $\overline{N_u}(c)$, is derived from the new (reduced) PRB utilization ratios fixed on a cell basis, $\overline{PRB_{util}}(c) = 0.05$ (i.e., 5 %) $\forall$ $c$. For this purpose, a curve relating both quantities is built from trace data (i.e., with the ML scenario). Figure 3 illustrates the relationship between both variables observed in traces. Each point reflects the hourly average of both indicators in 1 of the 129 cells in the scenario. From the figure, it is concluded that $\overline{PRB_{util}}(c)$ and $\overline{N_u}(c)$ are highly correlated. Specifically, the regression equation is

$$\overline{N_u}(c) = 2 \cdot 10^{-6}\overline{PRB_{util}}(c)^4 - 0.0002\overline{PRB_{util}}(c)^3 + 0.0071\overline{PRB_{util}}(c)^2 - 0.0746\overline{PRB_{util}}(c) + 1.2823\,, \quad (19)$$

with a determination coefficient of $R^2 = 0.71$. Note that such a correlation, obtained by aggregating all services per cell, is observed even when the traffic mix in all cells is not the same .

### 4.2. Results

Table 7 summarizes the main performance indicators obtained by RSM and RRSM scheduling models in the 2 scenar-

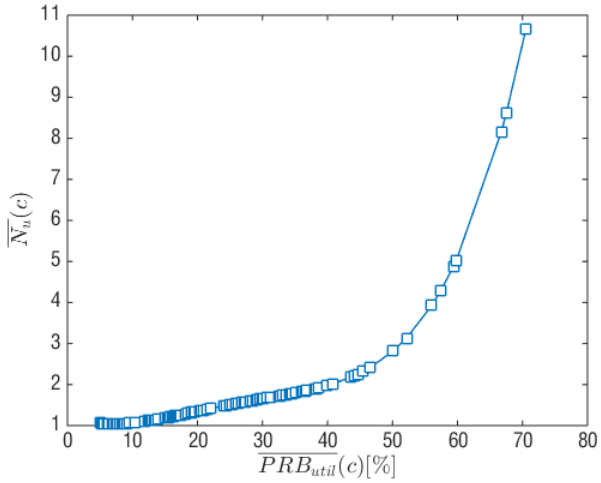



Figure 3: Relationship between PRB utilization ratio and number of simultaneous active users on a cell and hourly basis. $\overline{N_u}(c)$ vs $\overline{PRB_{util}}(c)$.

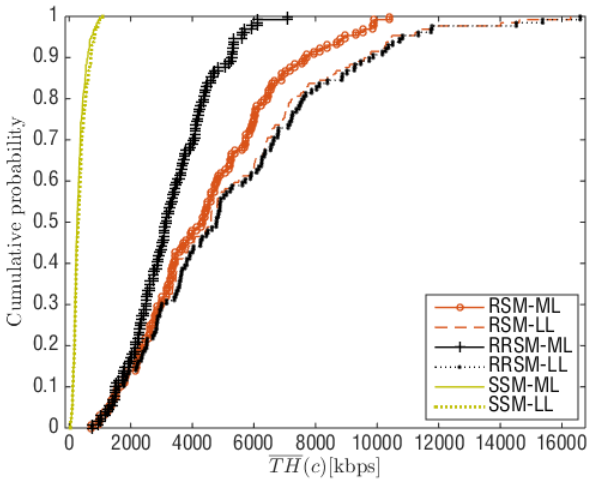



Figure 4: CDF for average cell throughput.

Table 7: Main indicators for RSM and RRSM scheduling performance models.

| Network load | ML | | LL | |
|---|---|---|---|---|
| Model | RSM | RRSM | RSM | RRSM |
| $\overline{\overline{N_u}}$ | 1 | 1.72 | 1 | 1.04 |
| $\max_c(\overline{N_u}(c))$ | 1 | 10.66 | 1 | 1.04 |
| $\overline{\overline{N_{PRB}}}$ | 50 | 36.57 | 50 | 47,79 |
| $\max_u(\overline{N_{PRB}}(u))$ | 50 | 47,79 | 50 | 47,79 |
| $\overline{\overline{TH}}[Mbps]$ | 25.72 | 18.81 | 34.31 | 32.71 |
| $\max_u(\overline{TH}(u))$ [Mbps] | 45.71 | 43.78 | 45.71 | 43.78 |

ios. The following indicators are included:

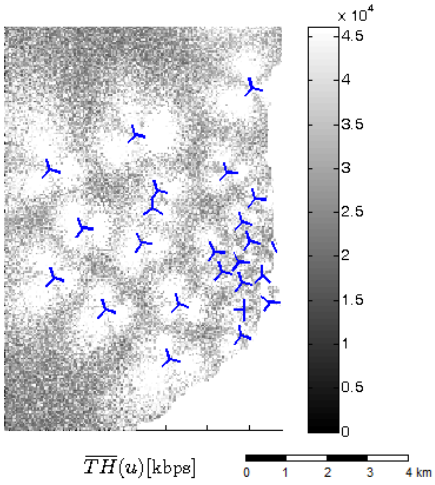
1. Global average number of simultaneous active users in the scheduler, $\overline{\overline{N_u}} = \text{avg}_c(\overline{N_u}(c))$.
2. Maximum number of active simultaneous users in the scheduler, $\max_c(\overline{N_u}(c))$.
3. Global average number of PRBs allocated per user, $\overline{\overline{N_{PRB}}} = \text{avg}_u(\overline{N_{PRB}}(u))$.
4. Maximum number of PRBs allocated per user, $\max_u(\overline{N_{PRB}}(u))$.
5. Global average user throughput, $\overline{\overline{TH}} = \text{avg}_u(\overline{TH}(u))$.
6. Maximum user throughput, $\max_u(\overline{TH}(u))$.

Is should be pointed out that, in the above-described indicators, a single overline (e.g., $\overline{N_u}(c)$) refers to the average operation across time (i.e., across TTIs), whereas double overline (e.g., $\overline{\overline{N_u}}$) indicates the average operation across time and cells/users. Moreover, the average operation across cells is a weighted average, so that cells with more users prevail over the others.

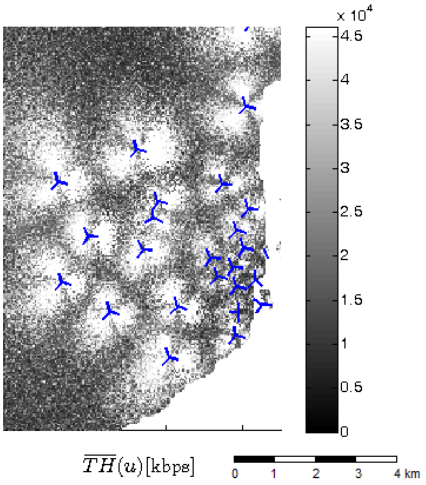
The analysis is first focused on the differences between the reference model, RSM (no scheduling, normal TTI), and the intermediate model, RRSM (RR scheduler, normal TTI). In Table 7, it is observed that RSM assumes that the number of active users in the scheduler is 1, so that the user is assigned the full system bandwidth (50 PRBs), leading to large values of user throughput. In contrast, RRSM takes into account that the number of simultaneous users is larger than 1, causing that the average number of PRBs assigned to the user is reduced by 27% in the ML scenario (from 50 to 36.57 PRBs) and 4% in the LL scenario (i.e., from 50 to 47.79 PRBs) on average. This leads to more realistic throughput estimates, especially for the ML scenario, where the average number of simultaneous users is larger (1.72). Specifically, $\overline{\overline{TH}}$ decreases from 25.72 to 18.81 Mbps (27% reduction) in the ML scenario, and from 34.31 to 32.71 Mbps (5% reduction) in the LL scenario.

For a more detailed analysis, Figures 5 (a) and (b) show the spatial user throughput ($\overline{TH}(u)$) distribution estimated by RSM in the ML and LL scenarios, respectively. The largest $\overline{TH}(u)$ values are obtained in locations close to the antenna and served by cells with low load (i.e., high spectral efficiency and high resource availability). The maximum value of 45.71 Mbps is close to the maximum theoretical bound obtained with a 10-MHz system bandwidth and a MIMO 2x2 configuration (≈50 PRB·1 Mbps/PRB) [36]. Figures 5 (c) and (d) show the same indicator for RRSM model. A similar spatial pattern is observed, but with lower throughput values as a result of the smaller number of PRBs assigned to the user.

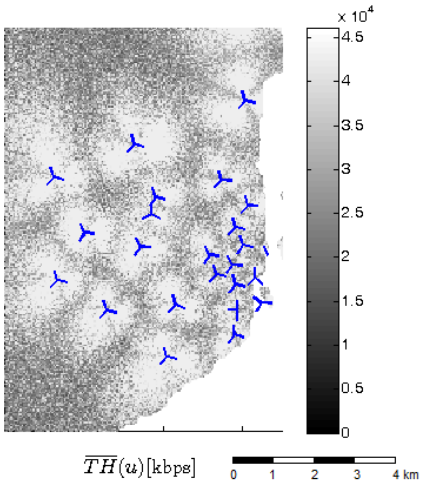
Unlike RSM or RRSM, SSM breaks down performance estimates on a service basis (e.g., $\overline{\overline{N_{PRB}}}(s)$). Table 8 presents the values of the main performance indicators obtained with the SSM performance model. It is observed that SSM tends to give lower estimates of $\overline{\overline{N_{PRB}}}(s)$ and, hence, of $\overline{\overline{TH}}(s)$. Specifically, in the ML scenario, $\overline{\overline{N_{PRB}}}(s)$ =1.33, 12.4 and 5.82 for Social/Web, App download and Video with SSM, respectively, versus $\overline{\overline{N_{PRB}}} = 36.57$ for all services with RRSM. This is due to the fact that SSM takes into account the last TTI effect on the average number of PRBs assigned to the connection. As expected, the difference is larger for services with more last



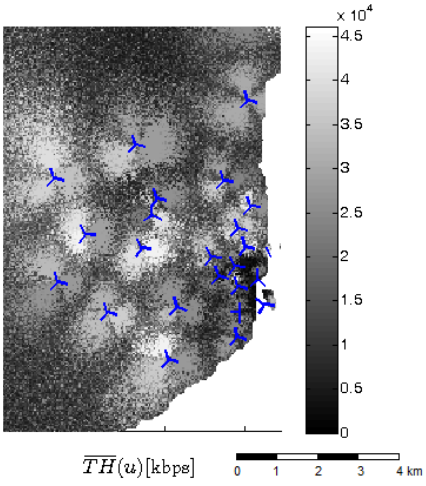

(a) RSM (LL scenario)




(b) RSM (ML scenario)




(c) RRSM (LL scenario)



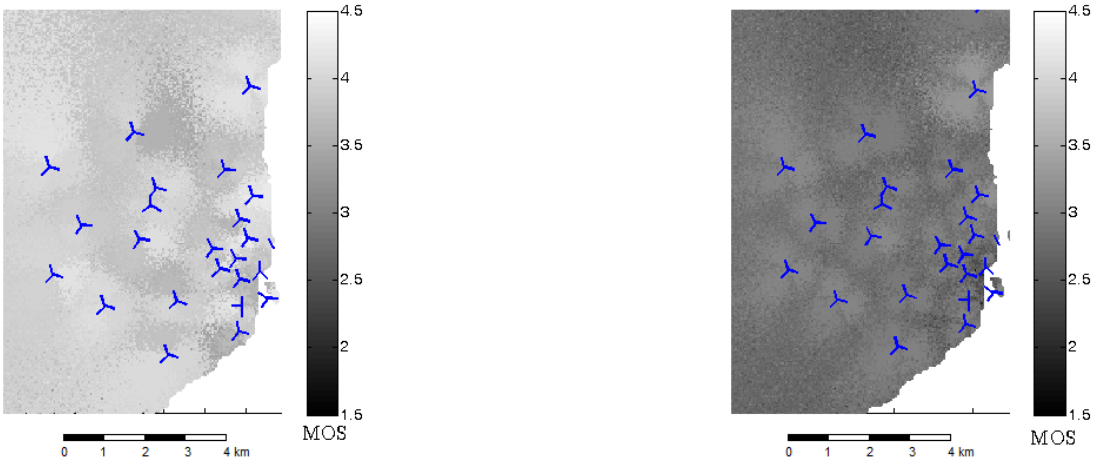

(d) RRSM (ML scenario)

Figure 5: User throughput map with RSM and RRSM models in ML and LL scenarios.

TTIs than normal TTIs (i.e., Social/Web, whose last TTI ratio is 0.995, as shown in Table 6). In contrast, the difference is lower in services with less last TTIs (i.e., App download, whose last TTI ratio is 0.742). Even for the latter, throughput estimates with SSM are lower than with RRSM. Thus, it is expected that SSM provides values closer to session throughput (which is the key indicator for QoE).

Also important, SSM suggests that service performance is very close in both network load scenarios for all services. This is counterintuitive, since the number of simultaneous users in both scenarios differs significantly ($\overline{\overline{N_u}} = 1.72$ and 1.04 in ML and LL, respectively). From this observation, it can be inferred that, even in ML, many active users are still in their last TTI, and therefore have a limited impact on normal TTI users. This is especially true for services with a large last TTI ratio (i.e., Social/Web or Video), where last TTIs are the most common situation for transmitting data. This result evidences the need for including the last TTI effect in scheduler performance models.

Figure 4 shows the cumulative distribution functions of cell throughput, $\overline{TH}(c)$, for the 3 scheduling models and 2 scenarios. Each point in the curves represents 1 of the 129 cells in the scenario. For clarity, a single curve is generated for SSM as the weighted average of all the services, so that services with more connections in the live network prevail over the others. Again, SSM shows lower average user throughput values compared to RSM and RRSM, for both ML and LL scenarios. In RSM and RRSM, the ML scenario has lower throughput values than the LL scenario. This is not the case for SSM, where ML and LL perform almost identical. These results again show that the most important factor for low throughput in low traffic networks is the large last TTI ratio and not cell load (as suggested in [37]), emphasizing the need for including the last TTI effect in the scheduling performance model. Consequently, RSM and RRSM throughput estimates can only be considered as an upper bound, only applicable to pure full buffer services.




(a) Social/Web service

(b) Video service

Figure 6: QoE maps derived with SSM.

Table 8: Main indicators for SSM scheduling performance model.

| Network load | | ML | LL |
|---|---|---|---|
| $\overline{\overline{N_u}}$ | | 1.72 | 1.04 |
| $\max_c(\overline{N_u}(c))$ | | 10.66 | 1.04 |
| $\overline{\overline{N_{PRB}}}(s)$ | Social/Web | 1.33 | 0.88 |
| | App download | 12.4 | 12.5 |
| | Video | 5.82 | 4.81 |
| $\max_u(\overline{N_{PRB}}(u,s))$ | Social/Web | 6.45 | 2.46 |
| | App download | 25.9 | 25.1 |
| | Video | 27.3 | 12.6 |
| $\overline{\overline{TH}}(s)$ [Mbps] | Social/Web | 0.54 | 0.56 |
| | App download | 7.67 | 8.5 |
| | Video | 2.59 | 3.16 |
| $\max_u(\overline{TH}(u,s))$ [Mbps] | Social/Web | 0.80 | 1.12 |
| | App download | 21.5 | 22.5 |
| | Video | 6.36 | 7.95 |

Table 9: Cell throughput figures traces vs. scheduler performance models.

| | Services | RSM | RRSM | SSM | Traces |
|---|---|---|---|---|---|
| $\overline{\overline{TH}}(s)$ [Mbps] | Social/Web | - | - | 0.54 | 0.6 |
| | App download | - | - | 7.67 | 16.05 |
| | Video | - | - | 2.59 | 1.15 |
| | Global | 25.72 | 18.81 | 1.87 | 2.17 |

For validation purposes, Table 9 compares the global average user throughput per service for the complete scenario estimated by SSM in the ML scenario against that reported in connection traces. A global weighted average throughput value (i.e., with no service differentiation) is also calculated, resulting in 25.72, 18.81, 1.87 and 2.17 Mbps for RSM, RRSM, SSM and trace data, respectively. Recall that the comparison on a service basis is only possible with SSM, since RSM or RRSM only provide global throughput figures with no service differentiation. It is observed that, for SSM, the difference compared to the real data is only 13%, whereas the difference for RSM and RRSM is 1200% and 900%, respectively.

To gain insight into the impact of cell load, the 3 models are tested in a highly loaded and an underutilized cell (denoted as C100 and C78, respectively). Specifically, $\overline{PRB_{util}}(C78) = 13\%$, while $\overline{PRB_{util}}(C100) = 59.8\%$. Table 10 shows a comparison of the different scheduler performance models in both cells, broken down per service when appropriate. The upper part of the table shows trace data, while the lower part shows estimates from the simulation tool with different performance models.

A preliminary analysis of trace data indicates that the higher cell load, $\overline{PRB_{util}}(c)$, in C100 is translated into a larger number of simultaneous users, $\overline{N_u}(c)$ (i.e., $\overline{N_u}(C78) = 1.15$ and $\overline{N_u}(C100) = 5.02$). This difference is due to a larger traffic demand, observed in $V^{DL}(c,s)$ and $N_{con}(c,s)$, since both cells have the same average spectral efficiency ($\approx$ 430 kbps/PRB). As in Table 6, it is observed that App download is the service with the smallest last TTI ratio in both cells. More interestingly, the last TTI ratio of all services is smaller in the highly loaded cell (C100), as a result of a lower user throughput due to less available radio resources per user in the cell. Such a decrease in the last TTI ratio is more evident for the service closest to the theoretical full buffer service (i.e., App download), whose $\overline{R_l}(c,s)$ decreases from 0.753 in C78 to 0.614 in C100. In contrast, in the service consisting of small data bursts (i.e., Social/Web), $R_l(c,s)$ only decreases from 0.998 in C78 to 0.988 in C100.

By comparing model estimates in both cells, it is observed

Table 10: Cell performance predicted by the proposed model in the ML scenario.

| **Cell** | | **100** | **78** |
|---|---|---|---|
| Input data (traces) | | | |
| $\overline{PRB_{util}}(c)$ [%] | | 59.8 | 0.13 |
| $\overline{SE}(c)$ [kbps/PRB] | | 431.7 | 436.9 |
| $\overline{N_u}(c)$ | | 5.02 | 1.15 |
| $N_{act}^{DL}(c)$ | | $2.33 \cdot 10^6$ | $5.22 \cdot 10^5$ |
| $V^{DL}(c,s)$ [MB] | Social/Web | $0.61 \cdot 10^2$ | $1.71 \cdot 10^1$ |
| | App download | $1.45 \cdot 10^3$ | $3.76 \cdot 10^2$ |
| | Video | $0.27 \cdot 10^3$ | $0.79 \cdot 10^2$ |
| $\overline{R_l}(c,s)$ | Social/Web | 0.9887 | 0.9985 |
| | App download | 0.6144 | 0.7526 |
| | Video | 0.8805 | 0.9353 |
| $\overline{R_n}(c,s)$ | Social/Web | 0.0112 | 0.0015 |
| | App download | 0.3855 | 0.2474 |
| | Video | 0.1195 | 0.0646 |
| $V_l^{DL}(c,s)$ [kB] | Social/Web | $6.18 \cdot 10^4$ | $1.57 \cdot 10^4$ |
| | App download | $9.15 \cdot 10^5$ | $4.38 \cdot 10^4$ |
| | Video | $2.42 \cdot 10^5$ | $4.12 \cdot 10^4$ |
| $N_{con}(c,s)$ | Social/Web | 10931 | 2349 |
| | App download | 1684 | 246 |
| | Video | 2375 | 362 |
| RSM | | | |
| $\overline{N_{PRB}}(c)$ | | 50 | 50 |
| RRSM | | | |
| $\overline{N_{PRB}}(c)$ | | 9.96 | 43.49 |
| SSM | | | |
| $\overline{N_{PRB,n}}(c)$ | Social/Web | 21.88 | 44.19 |
| | App download | 21.88 | 44.19 |
| | Video | 21.88 | 44.19 |
| $\overline{N_{PRB,l}}(c,s)$ | Social/Web | 1.582 | 1.701 |
| | App download | 4.478 | 7.014 |
| | Video | 2.576 | 3.506 |
| $\overline{N_{PRB}}(c,s)$ | Social/Web | 1.801 | 1.779 |
| | App download | 11.19 | 16.06 |
| | Video | 4.884 | 6.097 |
| $\overline{TH}(c,s)$ [Mbps] | Social/Web | 0.694 | 0.641 |
| | App download | 4.683 | 6.534 |
| | Video | 1.987 | 2.365 |

that RSM assigns the same number of PRBs per user ($\overline{N_{PRB}}(c) = 50$) in both C100 and C78, regardless of cell load. In contrast, RRSM assigns less PRBs to users of the highly loaded cell (C100), as it takes the number of simultaneous users per cell, $\overline{N_u}(c)$, into account. Specifically, $\overline{N_{PRB}}(C100)$ is 77% smaller than $\overline{N_{PRB}}(C78)$ (9.96 vs 43.49). In SSM, service differentiation results in a different number of PRBs per user across services. As expected, the number of PRBs per user in normal TTIs, $\overline{N_{PRB,n}}(c)$, is smaller in the highly loaded cell (44.19 in C78 vs 21.88 in C100), due to a larger number of simultaneous user. Note that, even if $N_u$ is 5 times larger in C100, $\overline{N_{PRB,n}}(c)$ in C100 is only half of that of C78. This is because some of them are last-TTI users, which do not compete for resources. By comparing $\overline{N_{PRB,l}}(c,s)$ in both cells, it is observed that all services also end up with less PRBs per user in last TTIs in the highly loaded cell (C100). However, such a decrease is only 7% for Social/Web (from 1.701 to 1.582) and 36% for App download (from 7.014 to 4.478). As a result, the total number of PRBs per user, $\overline{N_{PRB}}(c,s)$, in both cells is almost the same for Social/Web ($\approx 1.8$), but differs significantly for App download (i.e., 16.06 in C78 and 11.19 in C100). Such an assignment leads to the same user throughput in both cells for Social/Web (0.64 Mbps in C78 and 0.68 Mbps in C100), but very different throughput between cells for App download (6.536 Mbps in C78 vs 4.683 Mbps in C100, a 28% decrease). Thus, the model reflects that services consisting of small data bursts (Social/Web) are not affected by capacity limitations, whereas full-buffer services (App download) are degraded in the same situation. Video service, consisting of large data bursts sent periodically, is an intermediate case. Thus, cell load still has an impact on resources assigned per user (and user throughput), but smaller than in the App download service (from 6.097 to 4.884 PRBs, and from 2.365 to 1.987 Mbps, a 20% and 16% decrease, respectively).

Finally, Figure 6 shows QoE maps of the two services with the lowest and highest requirements, derived from throughput estimates obtained with SSM in the ML scenario. Note that RSM and RRSM do not differentiate between services, and cannot be used to build QoE maps on a service basis. It is observed that users of Social/Web services have better experience than Video users, even if the number of PRBs per user (and, hence, user throughput) is much lower.

## 5. Conclusions

In this work, a data-driven model for evaluating the performance of packet scheduling in a LTE grid-based system-level simulator has been presented. Unlike previous analytical approaches, the proposed model considers the effect of last TTI transmissions and can easily be adjusted to a real scenario using measurements in connection traces collected in the live network. Model assessment has been carried out with a real trace dataset, from which two different network load scenarios are generated.

Results have shown that considering the impact of last TTI transmissions is key for estimating session throughput accurately in radio planning tools, which can only be done on a service basis. This feature is of the utmost importance to evaluate the QoE of services that need less resources (e.g., instant

messaging, social networks ...). The impact of including last TTIs is significant, especially on the average amount of resources assigned to the user, with differences of up to 80%, and user throughput, with an average reduction of 82%, justifying the need for this feature. The proposed model can be integrated in radio network planning tools to check the impact of re-planning actions on end-user experience. It is especially suitable for big-data empowered SON platforms that make the most of network data (traces) [38]. For this purpose, throughput estimates computed on a service basis can be used to derive geolocated service performance indicators (e.g., average web page download time, number of Video stallings, initial Video playback time...), from which to obtain MOS figures per location. Ultimately, such a tool will ensure that additional network resources are deployed where they have the largest impact on user opinion.

## Acknowledgments

This work has been funded by the Spanish Ministry of Economy and Competitiveness (grant TEC2015-69982-R), the Spanish Ministry of Education, Culture and Sports (FPU grant FPU17/04286) and the Spanish Ministry of Science, Innovation and Universities (grant RTI2018-099148-B-100).

[1] Ramiro J, Hamied K. Self-organizing networks (SON): Self-planning, self-optimization and self-healing for GSM, UMTS and LTE. John Wiley & Sons; 2011.

[2] Chen CW, Chatzimisios P, Dagiuklas T, Atzori L. Multimedia quality of experience (QoE): current status and future requirements. John Wiley & Sons; 2015.

[3] Rupp M, Schwarz S, Taranetz M. The Vienna LTE-advanced simulators. Springer; 2016.

[4] Mehlführer C, Wrulich M, Ikuno JC, Bosanska D, Rupp M. Simulating the Long Term Evolution physical layer. In: 17th European Signal Processing Conference. IEEE; 2009, p. 1471–8.

[5] Olmos J, Serra A, Ruiz S, Garca-Lozano M, Gonzalez D. Link level simulator for LTE downlink. In: 7th European Meeting COST-2100. Pervasive Mobile & Ambient Wireless Communications. 2009,.

[6] González D, Ruiz S, García-Lozano M, Olmos J, Serra A. System level evaluation of LTE networks with semidistributed intercell interference coordination. In: IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications. 2009, p. 1497–501.

[7] Piro G, Baldo N, Miozzo M. An LTE Module for the Ns-3 Network Simulator. In: Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques. 2011, p. 415–22.

[8] Simsek M, Akbudak T, Zhao B, Czylwik A. An LTE-femtocell dynamic system level simulator. In: 2010 International ITG Workshop on Smart Antennas (WSA). 2010, p. 66–71.

[9] Muñoz P, de la Bandera I, Ruiz F, Luna-Ramírez S, Barco R, Toril M, et al. Computationally-efficient design of a dynamic system-level LTE simulator. International Journal of Electronics and Telecommunications 2011;57(3):347–58.

[10] Karabulut U, Awada A, Lobinger A, Viering I, Simsek M, Fettweis GP. Average downlink SINR model for 5G mmWave networks with analog beamforming. In: IEEE Wireless Communications and Networking Conference (WCNC). 2018, p. 1–6.

[11] Fernández-Segovia JA, Luna-Ramírez S, Toril M, Úbeda C. A fast self-planning approach for fractional uplink power control parameters in LTE networks. Mobile Information Systems 2016;2016:1–11.

[12] Piro G, Grieco LA, Boggia G, Capozzi F, Camarda P. Simulating LTE Cellular Systems: An Open-Source Framework. IEEE Transactions on Vehicular Technology 2011;60(2):498–513.

[13] Ruiz-Avilés JM, Luna-Ramírez S, Toril M, Ruiz F, de la Bandera I, Muñoz P, et al. Design of a computationally efficient dynamic system-level simulator for enterprise LTE femtocell scenarios. Journal of Electrical and Computer Engineering 2012;2012:1–14.

[14] Munoz P, de la Bandera I, Barco R, Ruiz F, Toril M, Luna-Ramírez S. Estimation of link-layer quality parameters in a system-level LTE simulator. In: 5th International Conference on Broadband and Biomedical Communications. 2010, p. 1–5.

[15] Ruiz-Avilés JM, Luna-Ramirez S, Toril M, Ruiz F, De la Bandera-Cascales I, Munoz-Luengo P. Analysis of load sharing techniques in enterprise LTE femtocells. In: IEEE Wireless Advanced. 2011, p. 195–200.

[16] Lopez-Perez D, Valcarce A, De La Roche G, Liu E, Zhang J. Access methods to WiMAX femtocells: A downlink system-level case study. In: 11th IEEE Singapore International Conference on Communication Systems. 2008, p. 1657–62.

[17] Lopez-Perez D, de la Roche G, Valcarce A, Juttner A, Zhang J. Interference avoidance and dynamic frequency planning for WiMAX femtocells networks. In: 11th IEEE Singapore International Conference on Communication Systems. 2008, p. 1579–84.

[18] Lopez-Perez D, Valcarce A, de la Roche G, Zhang J. OFDMA femtocells: A roadmap on interference avoidance. IEEE Communications Magazine 2009;47(9):41–8.

[19] Han K, Choi Y, Kim D, Na M, Choi S, Han K. Optimization of femtocell network configuration under interference constraints. In: 7th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. 2009, p. 1–7.

[20] Jiang D, Wang H, Malkamaki E, Tuomaala E. Principle and Performance of Semi-Persistent Scheduling for VoIP in LTE System. In: International Conference on Wireless Communications, Networking and Mobile Computing. 2007, p. 2861–4.

[21] Jesús Sánchez Sánchez J, Gómez G, Morales Jimenez D, T. Entrambasaguas J. Performance evaluation of OFDMA wireless systems using WM-SIM platform. In: Proceedings of the 4th ACM international workshop on mobility management and wireless access. 2006, p. 131–4.

[22] Mogensen P, Na W, Kovacs IZ, Frederiksen F, Pokhariyal A, Pedersen KI, et al. LTE Capacity Compared to the Shannon Bound. In: IEEE 65th Vehicular Technology Conference - VTC2007-Spring. 2007, p. 1234–8.

[23] Buenestado V, Toril M, Luna-Ramírez S, Ruiz-Avilés JM, Mendo A. Self-tuning of remote electrical tilts based on call traces for coverage and capacity optimization in LTE. IEEE Transactions on Vehicular Technology 2017;66(5):4315–26.

[24] 3rd Generation Parthnership Project . Policy and charging control architecture 2014;TS 23.203:version 12.6.0, Release 12.

[25] Baumgarten J, Eisenblätter A, Jansen T, Kürner T, Rose DM, Türke U. SON laboratory: A multi-technology radio network simulation environment for the SON analysis. In: 9th International Symposium on Wireless Communication Networks (ISWCS), IWSON Workshop. 2012,.

[26] Sánchez A, Acedo-Hernández R, Toril M, Luna-Ramírez S, Úbeda C. A trace data-based approach for an accurate estimation of precise utilization maps in LTE. Mobile Information Systems 2017;.

[27] Baldo N, Giupponi L, Mangues-Bafalluy J. Big Data Empowered Self Organized Networks. In: 20th European Wireless Conference. 2014, p. 1–8.

[28] 3rd Generation Parthnership Project . Subscriber and equipment trace: trace data definition and management 2014;TS 32.423:version 11.7.0, Release 11.

[29] Peng L, Yang B, Chen Y. Effective packet number for early stage internet traffic identification. Neurocomputing 2015;156:252 –67.

[30] Zhang J, Chen X, Xiang Y, Zhou W, Wu J. Robust Network Traffic Classification. IEEE/ACM Transactions on Networking 2015;23(4):1257–70.

[31] Gu C, Zhang S, Sun Y. Realtime Encrypted Traffic Identification using Machine Learning. JSW 2011;6(6):1009–16.

[32] Aggarwal CC, Reddy CK. Data clustering: algorithms and applications. CRC press; 2013.

[33] Oliver-Balsalobre P, Toril M, Luna-Ramírez S, Ruiz Avilés JM. Self-tuning of scheduling parameters for balancing the quality of experience among services in LTE. EURASIP Journal on Wireless Communications and Networking 2016;2016(1):7.

[34] Jiménez LR, Solera M, Toril M. A Network-Layer QoE Model for YouTube Live in Wireless Networks. IEEE Access 2019;7:70237–52.

[35] Huawei . Video Experience-based Bearer Network Technical White Paper. https://www.huawei.com/media/CORPORATE/PDF/white%20paper/video-

experience-based-bearer-network-technical-whitepaper;
accessed on Mar. 2019.
[36] Sesia S, Baker M, Toufik I. LTE-the UMTS long term evolution: from theory to practice. John Wiley & Sons; 2011.
[37] Buenestado V, Ruiz-Avilés JM, Toril M, Luna-Ramírez S, Mendo A. Analysis of throughput performance statistics for benchmarking lte networks. IEEE Communications Letters 2014;18(9):1607–10.
[38] Imran A, Zoha A, Abu-Dayya A. Challenges in 5G: how to empower SON with big data for enabling 5G. IEEE Network 2014;28(6):27–33.