Thesis #12 Report for CDSF (e-Horizon project)

Imane Oussakel, Philippe Owezarski, Pascal Berthou

▶ **To cite this version:**

Imane Oussakel, Philippe Owezarski, Pascal Berthou. Thesis #12 Report for CDSF (e-Horizon project). Rapport LAAS n° 20009. 2020. hal-02447058

HAL Id: hal-02447058
https://laas.hal.science/hal-02447058

Submitted on 21 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thesis 12 Report for CDSF

January 2020

Phd student: Imane Oussakel
Supervisors: Philippe Owezarski, Pascal Berthou

Contents

1 Introduction

With the tremendous growth of services and applications highly uploading data as much as downloading it, such as video conferencing and Virtual Reality (VR) streaming, the uplink throughput monitoring is required. However, throughput is considered as a crucial QoS metric in cellular networks. In fact, the chaotic radio phenomena characterizing the real environment lead to a degradation for the different QoS metrics, especially the throughput. Such phenomena include noise, multipath fading, and interference, etc. It is increasingly hard to understand how the throughput changes rapidly depending on the environmental situations. Therefore, radio phenomena should be considered when monitoring the uplink throughput in such environments. As recent researchers endorse the use of 4G standard for the short and middle term services/applications, the uplink throughput monitoring in such networks is then considered.

2 Technical Objectives

We propose in this work to turn our focus on enhancing uplink transmissions through monitoring its QoS. Such objective might be realized either with a reactive or proactive systems. The former indicates the system reaction after receiving an event. For example, the network congestion control mechanisms are triggered only after detecting a QoS degradation due to the congestion. On the other hand, pro-active systems anticipate any positive or negative variation in the system and activate the appropriate procedures to avoid any QoS deterioration. It is clear that the QoS is maintained better in proactive systems compared to the reactive ones. Therefore, the uplink QoS monitoring in a proactive context is considered. For that, we propose to investigate the predictability of uplink throughput in 4G networks.

In such networks, the lower layers are the ones responsible for error correctness and treating any signal deformation due to the wireless channel conditions. They play a pivotal role in the throughput determination. Therefore, we investigate the throughput estimation based on the eNB lower layer metrics.

3 State of Art

3.1 Throughput estimation

Over years, throughput estimation and prediction has been widely studied in wired networks and WLAN. D. Koutsonikolas et al. [1] reveal the ineffectiveness of using those techniques in cellular networks as they are characterized by the large short-scale fluctuation of bandwidth. Nevertheless, [3], [4] and [2] prove the predictability of throughput in cellular networks. Consequently, only the proposed techniques in that environment are discussed. They can be categorized into two categories, as shown below. The approaches estimating the data-rate based on higher layer metrics such as TCP metrics, and the ones considering the lower layer metrics (i.e. physical and MAC (Medium Access Control) layers).

3.1.1 Estimation based higher layer metrics

The higher layer metrics, starting from the network layer up to the application layer are used to estimate the throughput, mainly for one upcoming time interval. For instance, Winstein et al. [6] propose the use of packet inter-arrival time to infer link bandwidth and further determine the number of packets that can be transmitted. Q. Xu et al. [5] estimate the throughput based on the historical throughput and the instantaneous sending rate, using regression trees. The developed protocol monitors the network traffic passively. Hence, it limits the possibility of predicting the achievable throughput of the following time window at an arbitrary sending rate. It assumes also that the sending rate of the upcoming time window is similar to the previous one. Such presumption might be limited in network with high throughput variability.

The proposed techniques in these studies focus only on the higher layer metrics, neglecting the rich lower-layer information.

3.1.2 Estimation based on lower layer metrics

Considering the fact that data-rate performance is affected by radio phenomena, and lower layer metrics reflect the channel conditions, several studies have investigated the throughput estimation based on lower layer metrics for different use cases. For instance, to improve the TCP cross-layer congestion control mechanism in 3G networks, F.Lu et al. [2] propose a prediction of downlink capacity based on CQI (Channel quality indicator) and DRX (Discontinuous Transmission). However, it uses a basic CQI-rate matching and the presence or not of DRX during a given time interval to predict the link

capacity for the upcoming time interval. On the other hand, Margolies et al. [7] develop another version of the proportional fair scheduler, adapted for mobile users. The developed scheduler takes the predicted feasible data-rate as an input argument. Such prediction is based on the reproducibility of signal quality over the same path and user trajectory tracking. Simulations results are promising as the throughput has increased by 15%–55% compared to the traditional schedulers, while improving fairness.

On the other hand, with the emergence of the machine learning techniques and its good performance in different application domains, recent studies estimate the throughput with an underlying machine learning algorithm. It includes mainly Support vector machine (SVM) and Random forest (RF). For instance, in order to minimize the energy consumption by cellular UEs, A.Chakraborty et al. [8] propose a protocol based on a specific LoadSense technique to increase the UE coordination efficiency for transmissions. The LoadSence approach is based on support vector machine classifier. It uses features such as the link quality, power ratio and its variation to estimate the availability of low or high throughput for the UE. The throughput was measured by downloading data from a server located in the same geographic area as the UE. In this study, it is considered that any throughput variation is mainly due to the wireless channel contention and not the operator wired part. In fact, this might be a misleading hypothesis as the authors had no control of the operators side.

Then, with the objective to help content provider choose the most appropriate representation (e.g. picture resolution, video resolution and rate) before the connection establishment, A. Samba et al. [15] consider the throughput estimation before establishing the connection. For that, they conduct a measurement campaign involving 60 users connected to a production network in France. They have proven that using either radio measurements on the UE side or the RAN measurements (e.g. average cell throughput, average number of connected users, BLER (Block Error Rate) of the cell) lead to good estimations. The application of random forest on UE radio measurements, RAN metrics and their combination result respectively in 50%, 59% and 65% of the relative prediction error within $\pm 20\%$. The relative prediction error of an estimation is computed by dividing the estimated value minus the actual value, by the actual value. Further, C Yue et al. [20] developed a machine learning based framework to estimate the UE average throughput, named bandwidth. A server sends an UDP traffic to the UEs in different scenarios, i.e. stationary and mobility scenarios. An extensive measurements campaign is then conducted in two commercial LTE networks in the US. As a result, five lower-layer measurements are identified as the most correlated with the downlink bandwidth. The measures include RSRP

(Reference signal Receive Power), RSRQ (Reference Signal Receive Quality), CQI (Channel Quality Indicator), BLER, and the number of handover attempts. Based on the collected metrics and historical bandwidth, accurate predictions were obtained. For instance, for the walking scenario, 69% of the relative prediction errors are within ±10%. Overall, when using the machine learning techniques, mainly the average throughput is considered, as the instantaneous throughput in [15] is estimated before the connection establishment.

Although, both client and network throughput are important in cellular networks, link bandwidth estimation related work considered only downlink transmissions. Our work differs in that we are interested in UL that takes advantage of a different transmitter and receiver composition. On the other hand, in order to estimate the uplink throughput an access to the eNB measurement in different radio scenarios is inevitable.

3.2 Test platforms

The aim is to monitor the UL QoS efficiently through investigating the uplink (UL) performance and its throughput estimation. It concerns mainly the observed QoS at the network side instead of UE side. For that, a need to access the network measurements, especially the eNB metrics is inevitable. However, the open access datasets such as [17] and [18] collect the measures/metrics mainly from the UE-side in real or simulated environments. The cellular networks are private and proper to the MNO (Mobile Network Operator). Hence, the eNB metrics and measurements are confidential. Taking such fact into consideration, researchers have developed simulators and SDR based platforms to test the cellular networks and integrate evolutionary services. In this part, the existing 4G simulators and SDR (Software Defined Radio) implementations are presented.

3.2.1 Simulations based solution

Even though the high demand for cellular network evaluation environments, a few open-source contributions have been proposed over the last decade. We differentiate mainly between the physical layer and high system level simulators. For instance, Mehlfuhrer et al. [10] have developed a mtalab based simulator for DL physical layer, proposing different capabilities such as one UE-one eNB, multi-users connected in one single cell and multi-user in a multi-cell environments. On the other hand, mainly four strong contributions could be considered for the high system level simulations, modeling almost the entire protocol stack of the 4G components. J.Ikunu et al. [11]

developed a matlab based simulator taking only downlink transmissions into account. LTE-sim proposed by Piro et al. [12] implements the main resources scheduling techniques for LTE network in multi-user/multi-cell environment. The later doesn't offer any support for the simulation workflow automation, e.g. definition and measures collection with a lack of one of the main eNB mechanisms, HARQ (Hybrid ARQ). Because of the physical layer complexity requiring a high computational effort, the aforementioned simulators implement an analytical model of the PHY layer, with no time notion, instead of a complete one.

Other scientists have developed the 4G protocol stack within the ns-3 framework [14] and simuLTE [13] based on omnet++. Although simuLTE [13] integrates many PHY layer mechanisms, it still remains incomplete as the physical channels are not modeled down to the OFDM symbols level. In other words, the later developed simulators are packet oriented based on discrete-event simulations. They either model the protocols layer or abstract them partially/ altogether.

3.2.2 SDR based solution

All of the above mentioned simulators are either implementing the LTE from a system level, abstracting some of the protocol layers, or only the physical layer. Thus, it turns the 4G network incomplete. Therefore, in order to compensate mainly the lack of the implementation of the physical layer in simulators, recently the Software Defined Radio (SDR) platforms are taking place. The SDR based platforms offer a high level of realism and flexibility due to the introduced softwarization. In fact, SDR [32] refers to the radio transceiver/receiver system implementing, in software, the traditional hardware component, i.e. amplifiers, filters, etc. To support the SDR capabilities, different boards are now developed by several companies such as USRP (Universal Software Radio Peripheral) by National Instrument [22] and limeSDR by MyriadRf [23].

Further, several SDR based platform implementing LTE have emerged recently. The platforms include LTE 100 which provides the use of eNB and EPC full network functionalities over a standard linux-based PC interfaced with USRP SDR platform. Unfortunately, it is not open source and commercialized by Amarisoft [24]. Also, other closed-source LTE based SDR implementation are available such as PicoSDR10 by Nutaq [25].

On the other hand, most of the open-source LTE plateforms are not complete. There is LibLTE [26] offering a pre-alpha development of SDR UEs and eNBs. OpenLTE [9] implementing the 3GPP LTE specifications involved in the downlink transmissions and reception. And, gr-LTE [27]

developing only the receiver part.

The only complete open-source SDR based system is Openairinterface [19]. It is a full software implementation of 4G network and highly realistic compared to the aforementioned simulators. However, for its unicity it is chosen for our testbed deployment.

4 Uncertainty, Scientific and Technical Locks to overcome

4.1 Testbed deployment

The testbed objective is to generate 4G real time traffic in different radio scenarios. The wireless system is then LTE-A. R2lab [28] is used as an underlying testbed, which offers a wireless network with multiple Software defined radio (SDR) nodes inside an anechoic room. The RF propagation inside the room is controlled thanks to the microwave absorbers materiel on the walls, scattering any wireless signal that comes across. The SDR based nodes allow a full access to realistic physical metrics/measurements. With remote access to the indoor nodes, the deployment of LTE-A network is accomplished. Fig. 1 schematizes the testbed. Openairinterface (OAI) software based platform [19] is implemented, which spans the full LTE 3GPP protocol stack (including features from LTE-Advanced and LTE-Advanced-Pro) for both radio access network (E-UTRAN) and core network (EPC). The openairinterface softmodem is connected with a hardware platform for SDR: USRP B210 (Universal Software Radio Peripheral) attached to two antennas, receiver and transmitter antennas. The USRP B210 is connected to a host computer to perform processing, and then connected to a PC running EPC network, and accessing the internet toward a distant server. The testbed works on frequency band 7 with 5 MHz bandwidth and uses FDD mode, which corresponds to the traditional and stable version of openairinterface platform with USRPB210. Further, as our objective is to observe the impact of radio phenomena on the bandwidth variation, the server and the intermediate PCs are well provisioned for not behaving as a bottleneck during communication.

To emulate 4G connected nodes, two commercial UEs are used and placed inside the anechoic room. The hardware metallic enclosure boxes scattered inside the room are considered as fixed multipath sources, which we take advantage of in investigating the impact of multipath phenomenon on bandwidth variation.

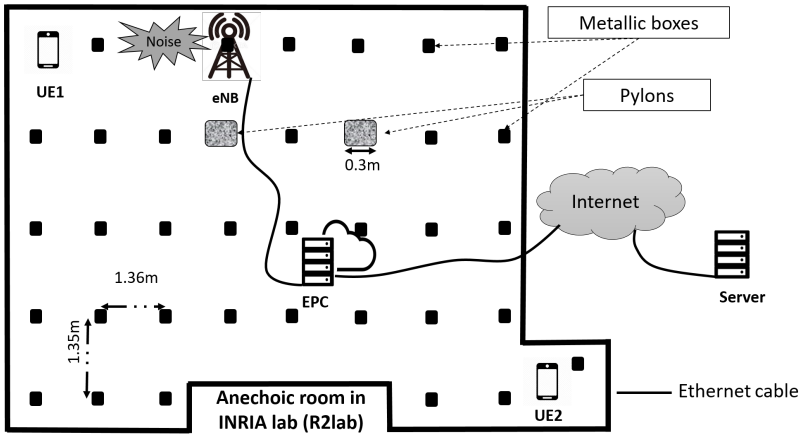

Figure 1: LTE-A Testbed deployment.

In real environments, multiple radio phenomena are scrambling the communications. Example phenomena include multipath fading leading to InterSymbol Interference (ISI) noise, pathloss and random processes such as AWGN (Additive White Gaussian Noise). These phenomena tend to attenuate aggressively the transmitted signal, which causes a significant amount of signal strength reduction and hence QoS degradation. The upcoming scenarios tend to investigate the impact of each phenomenon on the bandwidth prediction. For each test scenario, a radio phenomenon is added in the system to complexify the tests.

Scenario 1: Multipath fading For the first scenario, we investigate the impact of multipath fading and pathloss on bandwidth prediction/estimation. In other words, the two UEs used in the experimentation are on different distance from the eNB, i.e. each UE has a different pathloss. Over the room, multiple metallic boxes surround the eNB (fig. 1). They introduce multipath fading leading to ISI. The two pylons, covered with absorbers are considered as shadowers for the transmissions, especially for UE2. Therefore, only pathloss, shadowing and multipath fading are present in the system as scramblers for the transmission, i.e the anechoic room is isolated from any other radio phenomena. In order to avoid any degradation/losses due to the insufficiency of radio resources the total transmitted bandwidth by the two users is inferior to the maximum network capacity. In fact, after several tests the maximum uplink data-rate achieved in the testbed, by 1 connected UE, is around 8Mbps. Hence UE 1 transmits during the whole test duration a fixed amount of data of 2Mbps and UE 2 transmits 3Mbps. A benchmark of lower layers metrics is performed to construct datasets for ML algorithms as

explained further.

Scenario 2: Noise & Multipath fading In this scenario, another radio phenomenon is added in the anechoic room to investigate its presence on the bandwidth prediction, i.e, noise. For that we generate in a controlled manner a specific noise profile to scramble the communication.

Noise profile: In time varying scenarios, the received signal amplitude undergoes rapid fluctuation that is often modeled as a random variable with a particular distribution. For that we consider a Gaussian distribution, AWGN, which is characterized by its amplitude that affects the signal strength. Moreover, noise (AWGN) is introduced as it causes transmission errors and may disrupt the communication with ISI production for high power noise [33]. Contrarily to work in literature where AWGN is often taken with constant attenuation, we introduce randomness in the attenuation in order to have attenuation fluctuations of the signal over time. For that we define noise level as noise with a given gain and amplitude. We used mainly GNUradio on USRP B210. Given a list of gain levels and an interval of maximum and minimum amplitude levels, each 10 seconds a random value of amplitude and gain are chosen. In fact, the amplitude value affects the statistical characteristics of the noise source, i.e, the standard deviation of the Gaussian noise. The gain affects the transmitted signal power. The programmed step for noise level change (10 s) is chosen as to have sufficient samples for each noise level. Therefore, low noise level values keep the channel flat, while high noise level disrupt totally the communication, with the probability of introducing ISI, not only with noise but also with metallic boxes scattered in the room. Furthermore, the abrupt changes in noise levels during the transmission tend to reflect the real complex radio environments, where the user's mobility across different shadowers leads to aggressive/alleviated signal attenuation. The bandwidth is fixed to 5Mhz to scramble the full UL bandwidth. Both users transmit at same data-rate as in the scenario 1, and same benchmark of lower layer metrics is performed.

Scenario 3: Radio congestion & Noise & Multipath fading During the previous scenarios, the total transmitted bandwidth by the two UEs is much lower than the maximum available bandwidth. In this part, another radio phenomenon leading to bandwidth degradation is introduced in the testbed, i.e. radio congestion. The later occurs when the total bandwidth required by the connected UEs outpaces the maximum eNB bandwidth. Mobile application based speedtest is tested in scenario 1 for UE 1. It reaches the maximum throughput of 8 Mbps. Therefore, in order to realize radio

congestion, UE1 transmits its data at a fixed amount of 4 Mbps and UE2 at 5 Mbps. Noise is introduced as described in scenario 2 to complexify the test.

For all the above scenarios, IPERF3 generates traffic at the UE side and IPERF3-server monitors throughput reception in the server. In order to have a fixed transmission amount of data during the whole test duration, UDP (User Datagram Protocol) is used as a transport protocol. In fact, TCP (Transport Control Protocol) changes the transmission window based on the perceived packet losses in the window. Given this, any observed bandwidth degradation is essentially due to radio environment variation.

5 Field of exploration

Given the LTE-A testbed in the anechoic room, we are able to generate 4G traffic in different scenarios as described in the previous part. In this section, we present the methodology aiming at building our datasets.

5.1 Data collection

The eNB performs different measurements in order to decode the received data and adapt to channel variation. With SDR at the eNB side, we are able to collect all the performed eNB measurements and metrics, especially from the lower layers (physical and mac layer). As the connected UEs communicates their control signaling information using PUCCH, and transmit their data on PUSCH channels, we perform a deep benchmark of the main metrics/measurements linked with the two channels. The OAI eNB metrics/measurements are performed as depicted by the 3GPP standard. The collected metrics are mainly extracted during the lower layers data processing.

On one hand, radio measurements are collected as they are crucial for higher layer mechanisms and reflect the channel quality. The main measures are:

- Timing Advance (TA): As the propagation delay of different UEs is different based on their positions, TA is introduced to ensure synchronization between uplink and downlink at the eNB side. TA is a negative offset at the UE. Based on the UE transmitted PRACH, the eNB estimates the initial TA. Once the UE is connected, the eNB keep estimating TA and adjusting it by transmitting to the UE the required value, TA_update.

- SNR: SNR: Signal to Noise Ratio compares the level of the desired received signal to the level of noise. Taking P_{signal} and P_{noise} as the average power of the received signal and noise respectively, SNR is defined (in decibels) as follows: $SNR_{dB} = 10.log_{10}(P_{signal}/P_{noise})$. It is measured for each received PUSCH holding UE's data.

- received UL_CQI: Uplink Channel Quality Indicator. It is computed at the eNB based on the observed SNR.

- PUCCH received power and noise power: the two measures are estimated principally for each PUCCH handling a scheduling request.

- PUCCH threshold: It is the threshold to detect the pucch format1.

On the other hand, during decoding/demodulating the received channels, multiple metrics are extracted. In fact, when the UE is transmitting data via PUSCH, its data is transmitted using a Modulation Coding Scheme (MCS). The MCS is related to the modulation order Qm, e.g, QPSK (Quadrature Phase Shift Keying), 16 QAM (Quadrature Amplitude Modulation) and 64 QAM. The modulation scheme has a direct impact on the data-rate. However, when using a higher-order QAM scheme such as 64 QAM, each symbol represents a group of bits to be transmitted, i.e. each symbol has a 6-bit signature. The MCS is chosen based on the measurements exchange, reflecting the channel quality and the network capacity, between the eNB and the UE. Further, the transport block size (TBS) could be determined based on MCS value and the number of used resource blocks. TBS refers to data in the physical layer [32].

During the UE data transmission toward the network, eNB checks for data correctness using CRC (Cyclic Redundancy Check) and sends ACK/NACK to the UE. If the channel is good, errors are detected and corrected. On the other hand, when the channel quality is bad, the CRC might be insufficient. Further, LTE implements synchronous HARQ (contrarily to the asynchronous HARQ in DL) which combines ARQ error-control and a high-rate forward error-correcting coding (FEC). This way, the eNB requests a retransmission in case of error correctness inability. Such mechanism is characterized by multiple parameters, such as the round and redundancy version (rv). Is it possible to send ACK/NACK by the UE during the transmission, hence the length of ACK information in bits is considered. Moreover, the length of each received SDU/PDU (Service Data Unit/Protocol Data Unit) on the MAC layer is extracted with the buffer size when data is handled.

All of the aforementioned parameters are collected while decoding/demodulating the received PUSCH/PUCCH channels. In addition to other metrics usefull

for PUSCH and UL-SCH (UpLink Shared Channel) related functions, such as decoding time, decoding iteration, beta_offset, number of CRC bits and the cyclic shift. A total of 43 metric is collected during the whole test duration, i.e. 400 seconds.

One of our objectives is to compare the bandwidth predictions over different small time granularity. The bandwidth measurements are performed in a discrete time manner, each δt. As the minimum time report interval in IPERF3 is 100 ms, we fix $\delta t = 100ms$. The predictions are then made every δt. The eNB metrics are collected per subframe scale (1 ms). In order to have a representative measure per δt, we compute the maximum, minimum, mean, median and the standard deviation of each metric per δt to construct the datasets. Therefore, each metric γ is represented in the dataset as follows: $\{\gamma_{min}, \gamma_{max}, \gamma_{mean}, \gamma_{median}, \gamma_{std}\}$. For each UE u, $u \in \{1, 2\}$ and each scenario s, $s \in \{1, 2, 3\}$, a dataset is constructed, noted $dataset\_s\_u$. Each $dataset\_s\_u$ contains all the lower layers metrics collected for the given UE u during the scenario s, including historical received bandwidth.

5.2 Methodology

In this work, Python scikit-learn library is used for all the tests. GridsearchCV [29] is applied to choose the optimal hyper-parameters for each estimator. It combines both gridsearch and cross-validation methods. Gridsearch consists of an exhaustive search over subset values of parameters for a given estimator and cross-validation (CV) technique [34] estimates the prediction error of a model. CV is categorized into exhaustive and non-exhaustive categories, the former is more computational for high dimensional datasets. For that, the non-exhaustive cross-validation is chosen, mainly the recommended K-fold method, with K=10 to have a good compromise between variance and bias of the model [30]. Varma and Simon [31] report that the estimated prediction error from the cross-validation used to tune hyper-parameters is biased, and recommend the use of nested cross-validation instead, where an inner CV is used to select the optimized model (executed with GridsearchCV) and an outer CV to estimate the prediction error. Let denote K_1-fold and K_2-fold the inner and the outer CV respectively. Given an input dataset, a random split is performed to construct training and test sets. In fact, the dataset is split into K_2-folds, one fold is used for testing and the others K_2-1 folds constitute the training set. For each hyper-parameter combination from the gridsearch, K_1-fold is applied. It divides the training set into K_1 equal folds; K_1-1 folds are used for training and the remaining fold for evaluation. It computes the prediction error and iterates until all the folds are used for both training and validation, then the prediction

Algorithm 1 Bandwidth prediction from *dataset_s_u*

Input: `Historical metrics/measures with lag size` w, `historical received bandwidth, M: optimized prediction model.`

Output: $\hat{BW}$=$[(BW^1_{(1\delta t,w)},..,BW^1_{(i,w)}),..,(BW^m_{(1\delta t,w)},..,BW^m_{(i,w)})]$: `the predicted bandwidth.`

```
for each(w,i) do
    for t in range (1, m) do
        (BW^t_(1δt,w), BW^t_(2δt,w), .., BW^t_(i,w)))=M(X^t_test, X^{t-1}_test, .., X^{t-w}_test)
        X^t_test is the value of a feature in time t.
    end for
end for
return BW^
```

error is averaged over all the K_1 cases of CV. The hyper-parameter combination achieving a minimized prediction error is selected as the best optimized model. In order to generalize the selected model, the outer loop CV is used where the model is tested K_2 times on unseen data, i.e. the test set. Then, the generalized prediction error is the average of the estimated prediction error over the tested sets.

Algorithm 1 represents the prediction model for bandwidth. Given the dataset, we apply the algorithm summarized in the following. Let N be the size of the considered dataset. K_2-fold CV is chosen for having 25% of the dataset for testing which we denote X_{test}=$\{X^1_{test},..,X^m_{test}\}$ with size m, and 75% for the training set denoted X_{train}=$\{X^1_{train},..,X^n_{train}\}$ with size n, where X^p_{train} and X^p_{test} are the p^{th} samples in their corresponding datasets. GridsearshCV is applied on the X_{train} to select the optimized model, referred by M. Let w be the rolling window, it represents the past w time units, and i denotes the forecast window, where $i \in \{1\delta t...10\delta t\}$, with δt=100 ms. The maximum forecast window is then 1 second. For each lag w and forecast window i, M uses the historical metrics $(X^t_{test}, X^{t-1}_{test},..,X^{t-w}_{test})$ to predict for each sample X^t_{test} from X_{test} the corresponding bandwidth $BW^t_{(i,w)}$, $i \in \{1..10\}$. For example, with i=1 and w=2, M uses the current and two past time units of X^t_{test}: $[X^t_{test}, X^{t-1}_{test}, X^{t-2}_{test}]$ to predict the upcoming bandwidth in 100ms.

Therefore, for predictions evaluation, we compare the predicted bandwidth $BW^t_{(i,w)}$ with the received $BW^t_{(i,w)}$ bandwidth, based on RMSE (Root Mean Squared Error) metric. RMSE is attractive from a statistical and scientific perspective. It represents the average error prediction in the model, expressed in the units of the variable of interest. It is computed as follows: $RMSE = \sqrt{(}1/n * \sum_i^n (y_i - \hat{y}_i)^2)$ where $y_1...y_n$ are the actual values and $\hat{y_1}...\hat{y_n}$ the predicted ones. By squaring the error, a high weight is given to the large errors. RMSE score is negatively oriented, hence lower values are better. This part presents and discusses the results obtained on bandwidth prediction using machine learning techniques, i.e. LR and RF. Six datasets are tested with two ML techniques RF and LR, i.e. one dataset for each

UE per given scenario. The evaluation is based on RMSE as a performance metric.

5.3 Estimation accuracy

In order to compare the prediction performance of the two ML techniques, fig 2 exhibits error prediction based RMSE for each UE over all the scenarios. With the two MLs, the prediction error doesn't exceed 13 Kbytes. In general RF outperforms LR, except for dataset_1_1. For instance, for dataset_2_1 (scenario 2, UE1), the observed RMSE using RF is 6.44 Kbytes and 12.04 Kbytes using LR. It is to be noted that the maximum received bandwidth in scenario 2 is 82.3 Kbytes and 64 Kbytes for UE 1 and UE 2 respectively. Hence, RF leads to accurate predictions, around 7.8% of errors in an environment where only multipath and noise are present. For more complex scenarios such as scenario 3, the maximum received bandwidth is 92 Kbytes and 105 Kbytes for UE 1 and UE 2 respectively. The prediction error in such scenario reaches 13 %, but remain acceptable. Therefore, this results prove the possibility of predicting accurately uplink bandwidth using lower layer metrics/measurements in different radio scenarios.

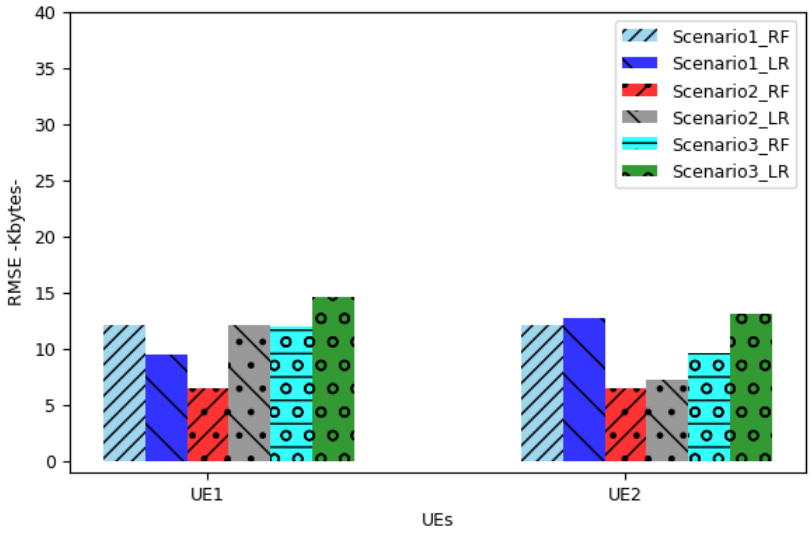

Figure 2: Error prediction for forecast window i=1δt.

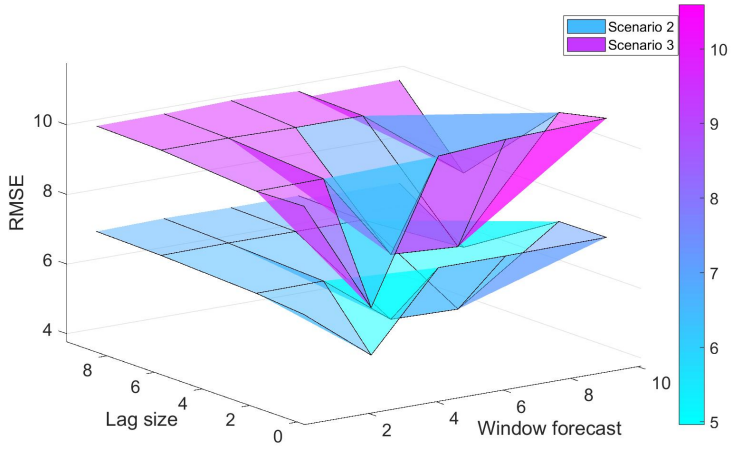

Figure 3: Sensitivity window forecast and lag size for UE 2.

5.3.1 Window forecast and lag size sensitivity

In order to exhibit the influence of window forecast size on predictions, we evaluate each dataset over $i\delta t$, with $i \in \{1, 3, 5, 7, 10\}$ and $w = 1$. That is, we predict bandwidth based on the instantaneous and the past lower layers measures/metrics, including the past received bandwidth. In addition, the importance of having numerous past radio measurements for good predictions is analyzed. The main lag sizes w tested are 3, 7 or 10, i.e. using the past three, seven and ten δt measurements. Fig. 3 plots in 3D the variation of the perceived RMSE over $i\delta t$ with $i \in \{1, 3, 5, 7, 10\}$ and per lag size w, $w \in \{1, 3, 7, 10\}$ for UE 2 prediction on dataset_2_2 and dataset_3_2 using RF as an underlying ML for predictions. Based on dataset dataset_2_2, the RMSE doesn't exceed 6.47 Kbytes for all the forecasted windows. The prediction at a granularity of $\delta t = 3$ leads to a smooth amelioration, i.e. the RMSE decreases by 1.49 Kbytes. Generally, when increasing the forecast window, a negligible increase of RMSE is observed based on both datasets. Thus exhibiting the insensitivity of the models to the forecast windows.

The introduction of lag size for predictions at granularity of $1\delta t$ doesn't improve necessarily the performance, i.e. using dataset_3_2, the RMSE increases from 6.46 Kbytes to 6.78 Kbytes when using $w = 10$. Such result points out the insensitivity to lag size when predicting at a granularity of $1\delta t$. On the other hand, a remarkable improvement is observed when predicting larger forecast windows and using lag sizes of $w = 3$ or $w = 7$. This indicates the sensitivity to lag size for predictions of higher forecast windows.

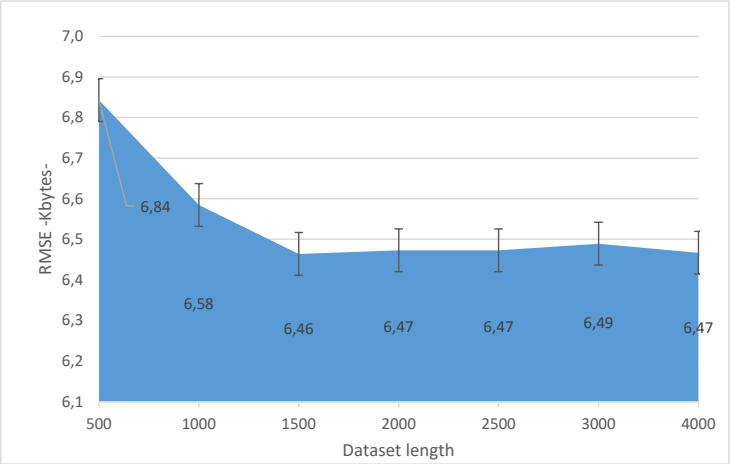

Figure 4: Sensitivity to training dataset length.

5.3.2 Sensitivity to training data length

Machine learning techniques leverage training sets to give accurate predictions. The size and variance of a given training set should then impact the prediction performance. In this part, we investigate the sensitivity of our model to the training dataset size. For that, we train RF model with different datasets lengths. In order to have a significant variance in each training dataset, we shuffle the main dataset, for instance dataset_2_2 with bandwidth dataset; 25 % is leaved out for test. At the beginning of the evaluation process, the dataset consists of 500 samples, i.e. training set length contains 375 samples. The size is then increased gradually for each test. Based on data_2_2, Fig. 4 depicts the prediction errors in terms of RMSE for all the tested sizes with $w = 1$ and $i = \delta t$. The main RMSE for all the tests is inferior to 6.84 Kbytes. The performance improves when increasing the dataset size from 500 to 1500 samples. Then, the prediction error remains between 6.46 Kbytes and 6.49 Kbytes for Larger training dataset sizes. This shows that the model becomes insensitive to training data length when it contains more than 1500 samples.

6 Retained solutions and acquired knowledge

In this work, we have investigated the instantaneous uplink throughput estimation in 4G network. For that a testbed based SDR is deployed in an anechoic room. Then radio phenomena are introduced to disturb the transmission channel. Machine learning techniques are used for the estimations. Accurate estimations are obtained.

In this work, we investigate the uplink bandwidth (throughput) estimation in cellular network and the radio phenomena impact on high-level QoS metrics, i.e. bandwidth. For that, a testbed is deployed in an anechoic room where the radio phenomena are controlled. This allows a clear analysis of encountered behaviors in bandwidth variation. Exhaustive benchmark of lower layers measurements/metrics is performed to constitute dataset reflecting the network response to radio environment variation. In order to predict the received bandwidth at a granularity of 100 ms, machine-learning techniques are used, mainly random forest and linear regression. Nested cross-validation is used for each case study to generalize the obtained error predictions. Different radio scenarios are tested, where the testbed complexity is increased gradually. For each scenario, the sensitivity of the models to forecast window, lag size and training data length are investigated. The model shows insensitivity when predicting. The two machine learning techniques lead to accurate estimations, especially random forest model. And, a strong correlation between throughput and radio environment variation is observed. The obtained results confirm the ability of the developed method to provide accurate uplink predictions. However, its application on real eNB measurements is prominent for its extension in real environments.

References

[1]] D. Koutsonikolas and Y. C. Hu. On the feasibility of bandwidth estimation in 1xEVDO networks. In Proc. of ACM Mobicom International Workshop on Mobile Internet Through Cellular Networks (MICNET), September 2009.

[2] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, and A. Terzis. CQIC: revisiting cross-layer congestion control for cellular networks. In Proc. of HotMobile, 2015.

[3] N. Abu-Ali, A.-E.M. Taha, M. Salah and H. Hassanein, "Uplink Scheduling in LTE and LTE-Advanced: Tutorial, Survey and Evaluation Framework," IEEE Communications Surveys & Tutorials, vol.16, no.3, pp.1239-1265, 2014.

[4] 3GPP 36.214. Evolved Universal Terrestrial Radio Access (E-UTRA), Physical layer Measurements (Release 15)

[5] Q. Xu, S. Mehrotra, Z. Mao, and J. Li, "PROTEUS: Network Performance Forecast for Real-time, Interactive Mobile Applications," in ACM MobiSys, 2013.

[6] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in Proc. USENIX Conf. Netw. Syst. Des. Implementation, 2013, pp. 459–472.

[7] R. Margolies, A. Sridharan, V. Aggarwal, R. Jana, N. K. Shankaranarayanan,V. Vaishampayan, and G. Zussman. Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms. In Proc. of IEEE INFOCOM, 2014.

[8] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee. Coordinating cellular background transfers using LoadSense. In Proc. of ACM MobiCom, pages 63–74, 2013.

[9] openLTE, [Online], Available: https://sourceforge.net/p/openlte/wiki/Home/

[10] C. Mehlfuhrer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the Long Term Evolution Physical Layer," in Proc. 17th European Signal Processing Conference EUSIPCO 2009, (Glasgow, Scotland), August 2009.

[11] J. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of LTE networks," in Proc. 71st Vehicular Technology Conference VTC2010-Spring, 2010.

[12] G. Piro, L. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework," IEEE Trans. Veh. Technol., vol. 60, no. 2, pp. 498 –513, Feb. 2011

[13] A. Virdis, G. Stea, G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE", DOI 10.1007/978-3-319-26470-7_5, in: Advances in Intelligent Systems and Computing, Vol 402, pp. 83-105, Springer, 15 January 2016

[14] N. Baldo, "The ns-3 LTE module by the LENA project," 2011

[15] A. Samba, Y. Busnel, A. Blanc, P. Dooze, and G. Simon. Instantaneous throughput prediction in cellular networks: Which information is needed? In IFIP/IEEE International Symposium on Integrated Network Management (IM), May 2017.

[16] C Yue, R Jin, K Suh, Y Qin, B Wang, W Wei, LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks- IEEE Transactions on Mobile Computing, 2017

[17] D. Raca, J.J. Quinlan, A.H. Zahran, C.J. Sreenan. "Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics" In Proceedings of ACM Multimedia Systems Conference (MMSys 2018), Amsterdam, The Netherlands, June 12 - 15, 2018

[18] MobileInsight dataset. Available on http://www.mobileinsight.net/download.html.

[19] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet, "Openairinterface: A flexible platform for 5g research," SIGCOMM Comput. Commun. Rev., vol. 44, no. 5, pp. 33–38, Oct. 2014.

[20] C Yue, R Jin, K Suh, Y Qin, B Wang, W Wei, LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks- IEEE Transactions on Mobile Computing, 2017

[21] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Doctor of Technology, Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.

[22] National Instrument SDR products. http://www.ni.com

[23] MyriadRF SDR components. https://myriadrf.org

[24] LTE 100 products by Amarisoft. https://www.amarisoft.com/technology/

[25] PicoSDR Series for Wireless Multi-Standard Prototyping https://www.nutaq.com/

[26] Tom Rondeau, 2014, GitHub repository,https://github.com/trondeau/libLTE

[27] Johannes Demel, Sebastian Koslowski, and Friedrich K. Jondral, A LTE Receiver Framework Using GNU Radio, Journal of Signal Processing Systems, 78(3):313–320, 2015.

[28] Thierry Parmentelat, Thierry Turletti, Walid Dabbous, Mohamed Naoufal Mahfoudi, Francesco Bronzino. nepi-ng: an efficient experiment control tool in R2lab. ACM WiNTECH 2018 - 12th ACM International Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization, Nov 2018, New Delhi, India. pp.1-8. ffhal-01857266f

[29] Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. Journal of the American Statistical Association, 70(350), 320-328. doi:10.2307/2285815

[30] J. Friedman, T. Hastie and R. Tibshirani, "The Elements of Statistical Learning", Springer Series in Statistics, 2001.

[31] S. Varma, R. Simon Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics, 7 (2006), p. 91

[32] www.3gpp.org: 3GPP Technical Specification 36.211, 36.213, 36.321.

[33] Bolat, E (2003). Study of OFDM Performance Over AWGNn Channels. B. Sc. Project, Department of Electrical and Electronic Engineering, Eastern Mediterranean University.

[34] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI, 1995.