Data-driven Evaluation of Anticipatory Networking in LTE Networks

Nicola Bui[1] and Joerg Widmer[2]

[1]Northeastern University, Boston, MA, USA
[2]IMDEA Networks Institute, Leganes (Madrid), Spain

***Abstract*—Anticipatory networking is a recent branch of network optimization that exploits contextual information to improve resource allocation decisions based on prediction. While some anticipatory networking concepts have been proposed in the literature, understanding of the potential real-world gains is so far very limited. Future mobile networks will likely integrate such mechanisms, and thus it is of paramount importance to understand the actual performance improvements and in which scenarios they can be realized. Analyzing a month-worth of LTE control channel information collected in four urban locations, we show how anticipatory networking can enhance current LTE networks. First, we propose a comprehensive optimization framework encompassing different forecasting solutions. Then, we provide a thorough analysis of the aggregated network traffic and the contributions of individual users. In particular, we show that predictable traffic accounts for more than 95% of the total traffic volume and that simple prediction and optimization techniques allow network operators to save 50% of the resources and/or on average more than double the offered data rate in our data set.**

***Index Terms*—LTE, Mobile, Sniffer.**

I. Introduction

Max Planck once said: "the assumption of an absolute determinism is the essential foundation of every scientific enquiry" [1]. Anticipatory Networking, a recent trend in network optimization, relates to the Nobel prize winner's aphorism by assuming that the future state of a dynamic system (i.e., the network) is, to some degree, predictable.

In fact, the fundamental principle of anticipatory networking is that the network performance can be improved by predicting the evolution of the system. Analyzing historical data and contextual information, it is possible to model [2]–[4] traffic dynamics at a cell-granularity and to profile user behavior at different time scales [5], [6], which can be used to control and optimize network operations.

According to the literature, anticipatory networking solutions can improve both the network efficiency in terms of spectrum utilization and enhance the Quality-of-Service (QoS) perceived by the users (see [7] and the references therein). For example, streaming applications can rely on buffered contents to avoid for using network resources when the signal quality is low and fill the buffer in the opposite situation. Knowing in advance whether the signal quality of a given user is going to improve or decrease allows the network to assign her resources when it is more efficient to do so and allows the user's application to modulate the amount of requested data according to the predicted achievable rate. There are many more applications that can be improved by anticipatory networking. In fact, the requirements of 5G communications will push the network efficiency to its limit and anticipatory networking is likely to become paramount to free up extra resources that will enable new applications.

The main missing element in the whole body of work on anticipatory networking is an in-depth evaluation of how predictive optimization would perform in the real-world. In this paper, we fill this gap by applying prediction-based optimization to the resource allocation data of LTE networks that we collected in four locations in Madrid over one month. In particular, we identify in the whole dataset those traces (i.e., almost continuous data flows belonging to a single user) that are suitable to be predicted. For these traces, we allow the data transfers to be re-organized so that future exchanges can be anticipated (i.e., buffered) if that improves a given objective function.

In particular, in our evaluation we treat all traffic that exhibits good predictability as elastic (i.e., it can be buffered in advance) and the rest as background traffic, which translates to a fixed and unpredictable load for the cell. This assumption allows us to study how the network would have performed, had it prediction capabilities. While not all predictable traffic is elastic, this is true for much of the high volume traffic such as video. Another important research direction (beyond the scope of this paper) is to make application traffic more elastic and provide means to signal delay requirements to the network. The main contribution of this paper is providing a thorough evaluation of anticipatory networking solutions using real-world data. In addition, we present the tools and the methodology we adopted to perform our evaluation and we present the datasets as well as their characteristics. Finally, our datasets are available on request for third parties to verify our conclusions and/or to perform their own tests.

Our analysis shows that omniscient optimizers can improve the average network efficiency by 35-40% in both communication directions, and more than double the data rate for downlink communication only (uplink data rate can be increase by circa 8% only, because of a consistently higher signal quality). The performance obtained using realistic predictors shows that anticipatory solutions are both feasible and effective, even though the performance is between 5 and 10% lower than the optimal. This confirms the preliminary results obtained in the literature over synthetic datasets and the benefit that predictive optimization can bring to next generation mobile networks.

In the rest of the paper, we discuss the following novel contributions. Section II illustrates the comprehensive anticipatory networking framework we use to evaluate the datasets. The section provides details about 1) time series prediction, 2) linear programming formulations to minimize network resources and maximize users' data rates, and 3) the complete optimization framework that encompasses prediction accuracy and objective functions. It also explains how to proceed from data collection to performance evaluation. Section III discusses our measurement campaign providing 1) a summary of the LTE characteristics, 2) a short description of the datasets, and 3) a preliminary analysis on the dataset where we distinguish the predictable (and thus optimizable) components from background traffic. Section IV examines the results obtained by the different anticipatory networking techniques on the datasets and provides further considerations about them and anticipatory networking in general. Finally Sections V and VI provide an analysis of the related work and our conclusions, respectively.

II. Anticipatory Optimization Framework

Anticipatory networking solutions include two main components: prediction and optimization. Here, we limit ourselves to a few selected methods that allow us to evaluate both the maximum achievable gains due to anticipatory networking and the improvements that realistic solutions would achieve in the real-world. We acknowledge that, depending on contextual information used and the application objectives, other solutions may exist that achieve different performance. However, our methodology proved to be adequate to solve our optimization problems in very large datasets and shed some light on the actual performance of anticipatory networking solutions. For a more detailed review of possible applications and variants of these components we refer the reader to [7].

A. Optimization Problem

We use [8] as a basis for our optimization problem, which is defined as a centralized decision making problem, where a set $\mathcal{N}$ of N users share a given quantity of network resources over a set $\mathcal{T}$ of T time slots, also referred to as optimization window. The objective of our formulation is to assign the available network resources so that all users obtain the requested information while the cost for the network is minimized. We use the following inputs for the problem:
Predicted achievable rate $r_{i,j} \in [0, r_M]$ is the prediction of the rate a user would achieve if no other user is scheduled. r_M is the maximum achievable data rate.
Requirement $d_{i,j} \in [0, q_M]$ is the minimum number of bytes needed in a given slot to stream the content at the minimum bitrate with no interruptions.

The problem is characterized by the following variables:
Resource assignment $a_{i,j} \in [0, 1]$ represents the average fraction of resources assigned to user i in slot j. In each slot, each user can be assigned at most the total available rate, $0 \leq a_{i,j} \leq 1$, and the sum cannot exceed the total available resources, $0 \leq \sum_{i \in \mathcal{N}} a_{i,j} \leq 1$.

Buffer state $b_{i,j} \in [0, b_M]$ tracks the amount of bytes stored in the buffer and b_M is the buffer size in bytes.
Outage $l_{i,j} \in [0, q_M]$ is the missing data to fulfill the minimum content requirement $d_{i,j}$:

$$l_{i,j} = [d_{i,j} - b_{i,j} - a_{i,j} r_{i,j}]_0^{d_{i,j}} \tag{1}$$

where $[x]_a^b = \min\{\max\{x, a\}, b\}$ is a bounding operator that forces the undelivered quantity to be greater than zero and smaller than the requirement in the slot.

In each slot j user i receives $a_{i,j} r_{i,j}$, which can be used either to satisfy the requirements in the current slot or to fill the buffer for later use. Thus we can write the following equation that describes the next buffer state:

$$b_{i,j+1} = b_{i,j} + a_{i,j} r_{i,j} - d_{i,j} + l_{i,j}. \tag{2}$$

We define $b_{i,0}$ as the initial status of the buffer of user i.

In addition, we introduce three metrics that we will use to build the objective function for our problem. Namely, we define the amount of used resources $\delta_i = \frac{1}{T} \sum_{k \in \mathcal{T}} a_{i,k}$, the fraction of continuous streaming time $\lambda_i = \frac{1}{T} \sum_{k \in \mathcal{T}} \left(1 - l_{i,k} d'_{i,k}\right)$ and the fraction of the extra quality obtained $\theta_i = \frac{1}{T} \sum_{k \in \mathcal{T}} \left(a_{i,k} r_{i,j} d'_{i,j} - 1\right)$, where we use $d'_{i,j} = 1/d_{i,j}$ if $d_{i,j} > 0$ and 0 otherwise to avoid division by zero.

Finally, we build two objective functions: the first minimizes the network resources spent, while the second maximizes the overall delivered data. Both objective functions must guarantee minimum outage before tackling the specific objective: if resources are not sufficient to satisfy the minimum requirements, both functions will give the same resulting allocation, which minimizes the overall outage. For the **resource minimization** we obtain the following LP formulation:

$$\begin{aligned} \underset{a,b,l}{\text{minimize}} \quad & \sum_{k \in \mathcal{N}} (\delta_k - K \lambda_k) \\ \text{subject to:} \quad & a_{i,j} \geq 0; \ \sum_{k \in \mathcal{N}} a_{k,j} \leq 1 - a_{B,j} \\ & l_{i,j} \geq 0; \ \ b_{i,j} \leq b_M \\ & l_{i,j} \geq d_{i,j} - a_{i,j} r_{i,j} - b_{i,j} \\ & \forall i \in \mathcal{N}; j \in \mathcal{T} \end{aligned} \tag{3}$$

where the weight K ensures that the solver's priority is on outage minimization and $a_{B,j}$ represents the fraction of resources used by background traffic at time j. We refer to the resources used for real-time or inelastic traffic, which cannot be moved and, thus, cannot be optimized as background traffic.

The **data rate maximization** LP is given by:

$$\begin{aligned} &\underset{a,b,l}{\text{maximize}} \sum_{k\in\mathcal{N}} (\theta_k + K\lambda_k) \qquad (4)\\ &\text{subject to:} a_{i,j} \geq 0;\ \sum_{k\in\mathcal{N}} a_{k,j} \leq 1 - a_{B,j}\\ &\qquad \sum_{k\in\mathcal{T}} a_{i,k} \leq a_{i,0};\ \ l_{i,j} \geq 0;\ \ b_{i,j} \leq b_M\\ &\qquad l_{i,j} \geq d_{i,j} - a_{i,j} r_{i,j} - b_{i,j}\\ &\qquad \forall i \in \mathcal{N}; j \in \mathcal{T} \end{aligned}$$

where $a_{i,0}$ is an upper limit to the total resources assigned to user i. Formally, the two optimization problems should have used mixed-integer formulations, because LTE resources are only assignable in finite quantities. However, since the time slots used for our optimization are two orders of magnitude longer than the LTE TTI, the expected approximation error is smaller than 1%.

B. Prediction Methodology

Among the many prediction techniques, we opt for time-series analysis, because it is simple to implement, to train and its computational complexity is sufficiently low. Here, we make no attempt to compare different prediction schemes and we do not claim the superiority of the methods used here, compared to other solutions. Our objective is to show a feasible solution that can be easily adopted in current networks. In addition, we evaluate the impact of prediction errors over the optimization quality.

According to previous optimization solutions [8]–[10], we need to predict users' achievable data rate, because by knowing the maximum data rate all users can be assigned at any given time allows to optimize the resource allocation process. Achievable rate is a function of the Modulation and Coding Scheme (MCS) obtained using standard LTE tables [11]. In addition, MCS expresses the spectrum efficiency of the communication and, thus, it is the best indicator of the cost/reward of communications. In our measurement campaign we collect and study MCS traces together with resources assigned to all the users and their achieved data rate.

We adopt AutoRegressive Integrative Moving Average (ARIMA) time-series analysis to model each of the traces and, subsequently, we use the obtained models to evaluate the prediction Mean Square Error (MSE). Since ARIMA models requires the time-series to have equidistant samples in time, before applying the model we regularize our traces: first, we analyze the average MCS over time bins and, then, we linearly interpolate our traces over gaps longer than one bin duration (i.e., when a given trace contains no information over a period longer than a bin). We fix the bin duration to 200 ms which allows reliable achievable rate estimation [12] while preserving the MCS variability induced by user movements. In addition, the selected bin duration should be long enough to filter fast MCS variation due to fast fading in most scenarios.

To verify the impact of linear interpolation over unknown gaps we test it over very dense traces collected with MobileInsight [13] and we create gaps to be filled by linear interpolation. Our tests shows that the error caused by linear interpolation is usually smaller than 5%, increasing substantially (max. 15%) only for long gaps and vehicular mobility.

An ARIMA model is characterized by three parameters: the autoregressive order p, the moving average order q and the degree of differencing d. For each of the traces, we choose the best orders for the ARIMA model according to the Box-Jenkins [14] methodology. This method, first, addresses non-stationarity by differencing the data if the autocorrelation plot has a very slow decay and, then, evaluates the times $p+1$ and $q+1$ when the autocorrelation and partial autocorrelation plots, respectively, become smaller than their 95% confidence interval. Then, we estimate the model coefficients by means of least square regression. Note that we create a model for each of the traces using all the information available for that trace. This allows to evaluate the best possible prediction obtainable with this methodology. In a real system, it might be impossible to have separate predictors per individual users and general models associated to user profiles might be used instead.

C. Evaluation Framework

In the previous parts of this section we defined our prediction and optimization tools. The reasons for our choices were mainly twofold: 1) test optimality (with perfect prediction and LP optimization) against suboptimal and more realistic options and 2) control the computational complexity to evaluate them on our dataset. In particular, we define the following features.

We include three levels of prediction accuracy:

- **Perfect:** the exact achievable rates are fed to the optimizer.
- **Proactive:** the prediction is computed by feeding the ARIMA models defined above with all the past samples of the trace. Since the optimizer can accurately know a given user achievable rate only when that user is actively using the medium, this type of prediction requires some sort of active achievable rate measurements when the user is not scheduled.
- **Reactive:** the prediction is still computed using the same ARIMA models, however, past information is only updated when the user is scheduled. To feed the optimizer with a continuous trace we fill the gaps by linear interpolation and we feedback the predictor output as it past input until a new scheduling event happens.

Note that, both the proactive and the reactive prediction types require to recompute both prediction and optimization at each time slot, in order to account for updated information.

We analyze two objective functions:

- **Resource Minimization:** we use the problem definition of Eq. 3 to compute the minimum amount of resources needed to provide each active user in the system with the same total rate they obtained in the original dataset. We enforce causality, by allowing users to use resources in the past to satisfy requirements in the future, but not vice versa.

- **Quality Maximization:** we use the problem definition of Eq. 4 to compute the maximum data rate that could be obtained by each active user in the system exploiting the same total quantity of resources. The parameter $a_{i,0}$ is set to match the original resource quantity consumed before the optimization.

In order to apply our evaluation framework on real data we proceed as follows:

1) Collect LTE scheduling information: we describe the tools we use and the locations where we perform the measurements.
2) Identify the predictable fraction of the traffic: active users exhibits characteristic features that help us distinguishing their trace from background/passive traffic.
3) Apply our evaluation framework on the obtained datasets.

III. LTE Measurements

We performed a one month measurement campaign in four LTE cells in Madrid. To collect the data, we used our Online Watcher for LTE (OWL) [15], a decoder of the LTE control channel. OWL uses a software-defined radio (SDR) to sample the LTE downlink channel and implements the decoding functionalities based on srsLTE [16], an open-source LTE library.

LTE scheduling measurements are possible because of centralized communication management and unencrypted control channel information. Centralized communications imply that a single base station, also known as eNodeB, coordinates the data transfers of the mobile phones, also known as user equipments (UEs), in both downlink and uplink channels. In particular, the eNodeB sends scheduling information to UEs using a dedicated channel. Thanks to our sniffer we are able to decode from the control channel the following information: 1) temporary user ID (C-RNTI) that does not allow to uniquely identify the user, but is sufficient to follow the scheduling of a given user over time until she stops her communications for longer than 10 seconds or she changes the cell, 2) assigned MCS, 3) allocated number of resource blocks, 4) transport block size. For space constraints, we refer the interested reader to [15] for further details.

A. Campaign description

Our measurement campaign consists of the data collected by OWL during one month in four different locations. We selected the four locations in order to analyze how optimization methods would performs in areas with different uses (e.g. residential, commercial, offices, education, etc.). In particular, we have been able to monitor two locations in Madrid and two in Leganes, a smaller town nearby. In the following, we will refer to them as *Callao*, *Rastro*, *Leganes* and *IMDEA*. Overall, we collected more than 100 GB of LTE scheduling information, corresponding to a total amount of 8860 terabytes of transferred data in the four locations.

The city locations in Madrid are close to the city center and they are characterized by a high density of commercial activity, while the locations in Leganes are more residential. Although all four locations include both pedestrian and vehicular users, their average speed in the city center is expected to be slower than that in Leganes. In all locations eNodeBs are placed on top of buildings of about four floors of height, but in Callao where the buildings are taller.

TABLE I
Dataset Statistics

	Callao	Rastro	IMDEA	Leganes
Operator	Movistar	Vodafone	Vodafone	Yoigo
Bandwidth	15 MHz	10 MHz	10 MHz	10 MHz
Frequency	1.8 GHz	800 MHz	800 MHz	1.8 GHz
Compressed Size	60 GB	19 GB	24 GB	4 GB
Total Time	35.5 days	37.5 days	21.3 days	18.7 days
Total Download	4.5 PB	0.86 PB	1.1 PB	0.15 PB
Total Upload	1.5 PB	0.3 PB	0.43 PB	0.02 PB
Total Traces	10.8 M	1 M	1.45 M	0.16 M
Active Traces	3.7 M	0.4 M	0.52 M	0.08 M
Med. D. Load	5 %	1 %	2.5 %	0.1 %
Med. D. Rate	1.13 Mbps	0.04 Mbps	0.24 Mbps	0.01 Mbps
Max D. Rate	21.3 Mbps	19.5 Mbps	22.2 Mbps	6 Mbps
A. Med. D. Rate	12.2 Mbps	12 Mbps	9.6 Mbps	14.1 Mbps
A. Max D. Rate	110 Mbps	75 Mbps	75 Mbps	75 Mbps
Med. U. Load	2.5 %	1 %	3 %	0.05 %
Med. U. Rate	0.36 Mbps	0.06 Mbps	0.16 Mbps	5 Kbps
Max U. Rate	18 Mbps	12 Mbps	12.3 Mbps	4.9 Mbps
A. Med. U. Rate	4.8 Mbps	2.7 Mbps	2.7 Mbps	2.3 Mbps
A. Max U. Rate	55 Mbps	37 Mbps	37 Mbps	37 Mbps

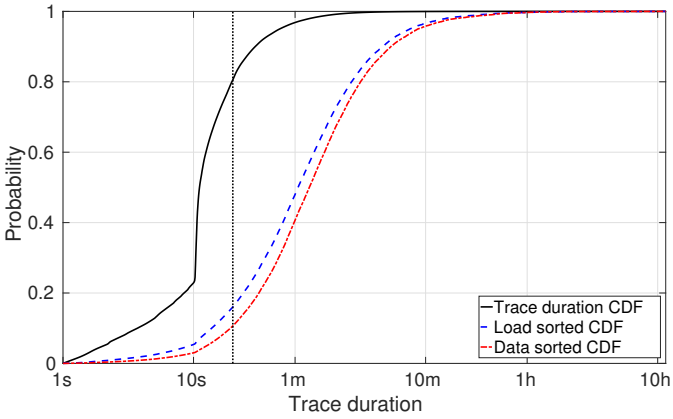


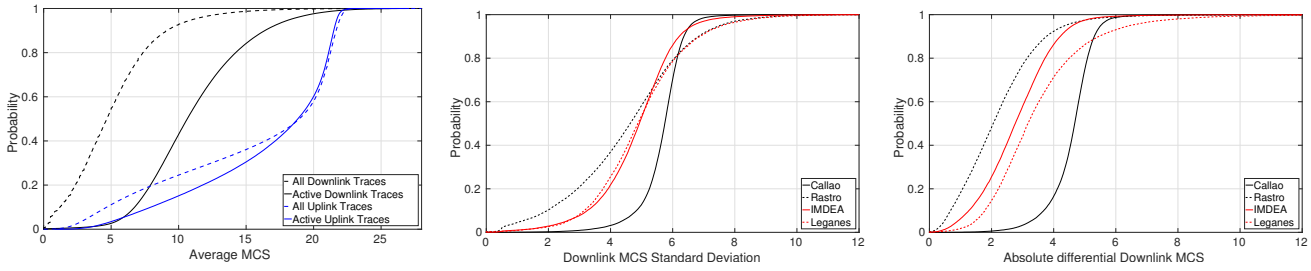
Fig. 1. CDF of the trace duration mapped to the sorted CDF of load and data rate in Callao.

Table I provides statistics information of the four datasets. Although all the locations show a low median load (< 5 %), in all of them the load averaged over 5 minutes reached peaks as high as 70 % of the available resources. The load is computed as the fraction of used resources over the available resources of the LTE channel.

B. Dataset Analysis

Since a user maintains her RNTI as long as she is active with no pause longer than 10 seconds, we split the traces accordingly: whenever a gap of 10 seconds or longer is present in a trace, it is split in two parts. Thus, we can analyze each trace in isolation and collect statistics about users network usage. In particular, each trace is a list of scheduling events concerning a particular user a containing:

- absolute time in milliseconds (LTE TTI)



(a) (b) (c)

Fig. 2. Fig (a) Compares MCS for active and all users for both downlink and uplink. The other two show downlink trace variability: Fig (b) shows the CDF of standard deviation of the MCS used in active traces; Fig (c) shows the absolute variation of the MCS.

- communication direction (downlink or uplink)
- MCS $\in [0, 31]$ (related to channel quality)
- N_{RB} (the number of resource blocks)
- transport block size (number of bits transferred)

For each collected trace we compute a set of compound metrics. The first three of them are trace duration, downlink trace size and uplink trace size. We first note that more than 60% of the collected traces are shorter than 10 seconds and are smaller than 10 kbit in terms of transferred data. This means that the majority of the collected traces carries little or no information. We assume that these *(*small) traces belong to background traffic performed by mobile phones without any active intervention from the user or it is related to automatic network management operations.

We analyze this in more details by computing the contribution to the total load of the traces longer than a given threshold or traces that transferred more than a given size of information. Figure 1 shows the trace duration CDF as a black solid line and maps the CDFs of the users' downlink load and total transferred size to their trace duration as dashed blue and dash-dotted red lines, respectively, for the Callao dataset.

The two CDFs represent the total load and data rate for all those users whose trace is longer than the value on the x-axis or, in other words, for a given duration on the x-axis, the three curves represent the fraction of traces shorter than that and the corresponding fractions of the total load and the total data transferred, respectively. Thus, traces shorter than 20s (dotted vertical line), which account for about 80% of the total traces (black line) constitute less than 20% of the total traffic (blue dashed line). A similar behavior can be found when analyzing the transferred size compared to the total load and it is valid for both downlink and uplink and for all the datasets.

Our next consideration is that short or small traces are not relevant to the objectives of anticipatory networking optimization. They are unlikely to provide Quality-of-Service (QoS) improvements, because they introduce little traffic and they are difficult to predict due to their short length and, thus, are difficult to be modeled. Additional evidence for this is obtained from the statistics of the average MCS measured over the traces.

Figure 2(a) shows the CDF of downlink (black) and uplink (blue) average MCS for all (dashed) and active (solid) users. Here we define a user to be active if its trace is either longer than 20 seconds or the transferred data size (either downlink or uplink) is larger than 100 Kbit. Note that this size corresponds to the size of a thumbnail image or that of a messaging application.

Both downlink and uplink CDF show that active users have higher average MCS, but also that downlink and uplink MCS distributions are quite different. The higher average MCS of active users is relevant for our analysis and shows that it is more likely for a user to be scheduled if she has a better signal quality, in case a larger volume of traffic is transmitted. However, the difference between downlink and uplink distributions, even though interesting, it is not directly relevant to the evaluation of anticipatory optimization. In fact, we believe they are mainly due to the specific cell topology and users' behavior in the area.

Now that we defined active users/traces and their contributions, we address cell aggregated results computed for all users compared to the contribution of active users only. A user's achievable rate is a function of the assigned MCS, which is, in turn, a function of the path loss (i.e., Channel Quality Indicator (CQI)) and the error probability. Before evaluating the performance of prediction techniques on the collected traces, we analyze the MCS statistics and their variation over time. In particular, we evaluate for each active user, the following metrics: average MCS, median MCS, MCS standard deviation, MCS range, standard deviation of the binned average MCS, average binned standard deviation of the MCS, average absolute variation of the binned MCS.

While the first four metrics are standard statistics obtained on the whole trace, the last three metrics are obtained by evaluating the traces over bins of equal duration: for each bin of a trace we computed the average MCS and its standard deviation. The overall idea is that the average MCS should be linked to the average path loss/signal quality experienced by the user, while the standard deviation should be linked to fast signal quality variation (i.e., fading). Thus, evaluating these metrics over the whole trace and over bins, we characterize traces in terms of signal quality, noisiness and their variation over time. Ideally, for a trace to be easily predictable, it should have a low noisiness and low quality variation. Figure 2 shows the CDF of the MCS standard deviation in the four datasets,

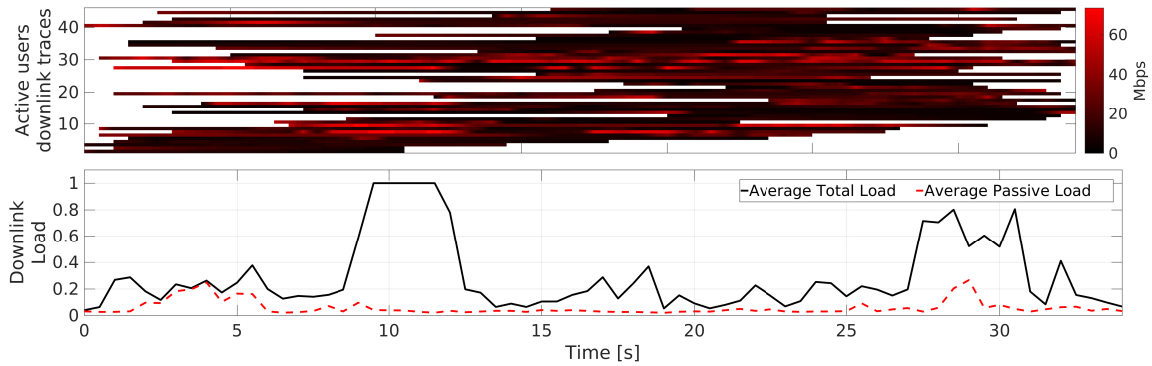


Fig. 3. A 35-second portion of the downlink channel of the Callao dataset. Each row of the top chart shows the MCS evolutions of an active user. The lower chart provide aggregated information of the cell traffic.

in the center, and the CDF of the average absolute variation of the binned MCS, on the right. In particular, Figure 2(b) shows that trace noise has a standard deviation usually smaller than 6 which means the range of MCS variation is small compared to the maximum range of 28. Also, the Callao dataset shows the highest noise, which can be a consequence of the particular topology of the area. Figure 2(c), which measures how fast the MCS varies in subsequent bins, tells us that the traces in the dataset have a slow to medium dynamic with successive MCS changes around 2-3 (max. range 28), which means that rapid large variations in MCS are not common.

IV. Evaluation and Discussion

In this section we investigate the performance of the different optimization approaches and degrees of prediction accuracy. To evaluate our framework, we proceed by selecting small portion of the datasets. Figure 3 provides an example of a 35-second analysis of the downlink channel, containing 45 active users. The top chart shows the evolution of the MCS for all the active users in the time frame, where each users is represented by a separate row and the color varies from white (no communication), to black (bad channel quality, few Kbps) fading into red (good channel quality, tens of Mbps). The bottom chart, instead shows aggregate information about the cell traffic: the average total load is shown as a solid black line and the contribution to the load generated by background traffic as a dashed red line.

Each portion of the dataset is generated as follows:

- select a subset of the dataset of length T and starting at time τ
- identify all N active users in the subset and retrieve their MCS traces
- create the ground truth elements $r_{i,j}$ from the MCS traces using the tables in the standard [11] to compute the transport block size for the maximum number of resource blocks. The ground truth is created for $i \in [1, N]$ and $j \in [\tau - \Delta_T, T + \Delta_T]$, where Δ_T is a margin to remove boundary effects from the evaluation.
- create ARIMA models and proactive predictions for all N users
- create minimum requirements $d_{i,j}$ and used resources $a_{i,0}$ as the amount of exchanged traffic and used resources, respectively
- create the background load $a_{B,j}$ for $j \in [\tau - \Delta_T, \tau + T + \Delta_T]$ summing the load of all non-active users
- run all the optimization schemes and compute their performance on the central time span $j \in [\tau, \tau + T]$. We refer to the resource allocation computed by the optimizer as $a^*_{i,j}$.

Thus, for each analyzed time span we obtain the *resource saving* percentage as

$$\Delta_a = \frac{100}{N} \sum_{i=1}^{N} \left(1 - \sum_{j=\tau}^{\tau+T} a^*_{i,j} / a_{i,0} \right), \tag{5}$$

the *data rate increase* percentage as

$$\Delta_r = \frac{100}{N} \sum_{i=1}^{N} \left(\sum_{j=\tau}^{\tau+T} a^*_{i,j} r_{i,j} / d_{i,0} - 1 \right) \tag{6}$$

and the *total outage* as

$$L = \sum_{i=1}^{N} \sum_{j=\tau}^{\tau+T} l^*_{i,j}. \tag{7}$$

In addition, in order to apply anticipatory networking optimization we assume that each active user's traffic can be re-organized and future data transfers can be buffered as soon as the trace starts, up to the maximum buffer size. According to a Cisco forecast [17] this assumption holds for 74% of the traffic in 2017 and is expected to grow in the next few years. When not specified otherwise, the buffer is assumed to be infinite.

After the initialization of one portion of the dataset and once all traces are organized in matrix form (one row per user, one column per time slot) we can solve the ideal optimization problems (Eqns. 3 and 4) using an LP solver, such as IBM CPLEX [18], to evaluate the optimal amount of resources needed (or the optimal quality of service level) together with their related outage. In order to account for the realistic

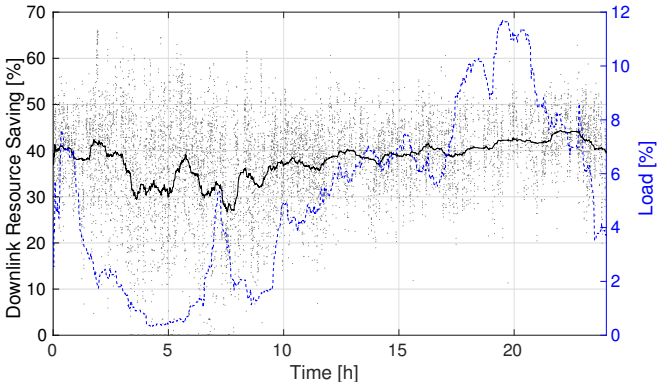


Fig. 4. Variation of the ideal optimizer performance over a full day downlink traffic in Callao, compared to the cell load. Lines illustrate the moving averages of the parameters, while dots are single results.

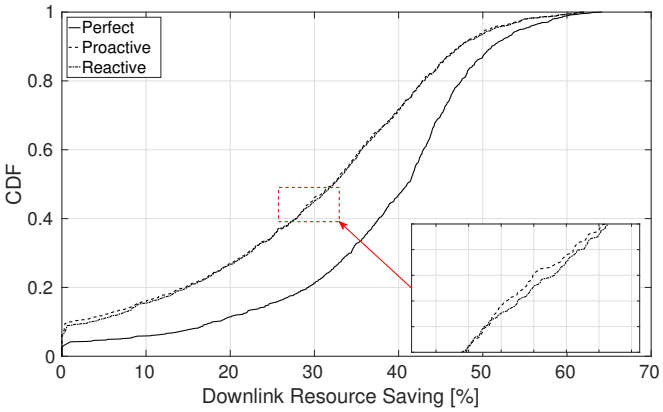


Fig. 5. CDFs of the resource saving obtained by anticipatory networking solutions for different prediction accuracies.

predictors we proceed iteratively over the trace matrix. At any given slot j we feed the solver with a smaller trace matrix obtained from the predictions computed with the data available up to time j. The proactive predictor computes the forecast using all the past samples of the traces, while the reactive predictor is fed with only those samples when resources are used and linearly interpolates between them. The results of the solver are considered as valid only for slot j and the process is repeated for the following slots until the end of the data set portion.

We start the result description with the performance of the ideal resource minimization optimizer with perfect future knowledge over a whole day. Figure 4 illustrates as a solid black line the average resource percentage saved over 30-minute moving windows. Grey dots represents single results computed over time spans of $T = 10$ and $\Delta_T = 5$ seconds. The blue dashed line illustrates the cell load variation averaged over 30-minute moving windows. The figure is obtained for the downlink channel of the Callao dataset.

First of all, the average performance of the resource minimization solution is very good. In fact, the solution is able to maintain an average saving almost always higher than 30% and up to 45%. However, the instantaneous performance of the solution is much more variable and spans the whole possible range from 0% (no improvement) to about 65%. These extreme conditions happen more frequently when the load of the cell is very low and, thus, they are symptoms of critical conditions in the analyzed portion of the dataset: such as a single active user whose trace is either already optimal (for 0%) or it allows for very high saving ($> 55\%$). For what concerns the impact of the cell load on the optimization performance, we cannot determine any strong correlation by visual inspection. We stress that a similar level of resource saving could not be achieved by aggressively filling the buffer as soon as possible: as shown in [9], the greedy strategy indiscriminately exploits time slots regardless of their quality (MCS). However, the range of individual results is wider for low load, while it gets smaller when the load is higher. When the cell load is higher, there are also more active users in the cell and, thus, the overall characteristic tends towards the average condition of the cell, while when the load is low, the individual behavior of each user dominates the aggregate characteristic of the cell traffic and determines the system performance.

Figure 5 shows the CDFs of the resource saving performance obtained by the three prediction accuracy levels (perfect, proactive and reactive).[1] The strongest impact on the system optimization is caused by replacing the perfect knowledge by more realistic approaches. Also, the chosen realistic approach does not strongly affect the amount of saved resource. A close inspection (see the zooms in the lower right part of the figures) allows to see the difference between the reactive and proactive predictions. Although they fare very similarly, the figures show that some higher resource savings are obtained by the reactive approach. This result might seem counter-intuitive, but is justified examining the other KPI: the outage time. In fact, while the proactive scheme never suffers from any outage, the average service outage of the reactive scheme is not zero in 3% of the cases only and never larger than 0.5 seconds. This is enough for the scheme to save some resources.

Overall the performance degradation due to realistic prediction methods ranges from 5-10% for high savings ($> 40\%$), to 10-15% for moderate savings (20-40%) to more than 15% for low savings. Even though this last condition happens in fewer than 15% of the analyzed cases, these are the cases where anticipatory networking is more likely to be useless or detrimental to the users' QoS: in fact, while some resources are still saved, they might be saved at the expenses of some outage, which will impact the users' experience.

Figure 6 shows the CDF of the quality maximization performance and is equivalent to the previous in all aspects, but for the magnitude of the improvements. In fact, the quality maximization solutions are able to more than double the data rate for the downlink channel. Conversely, in the uplink (not shown because of space constraints) the improvements

[1]Uplink charts are omitted, because they are very similar to the downlink.

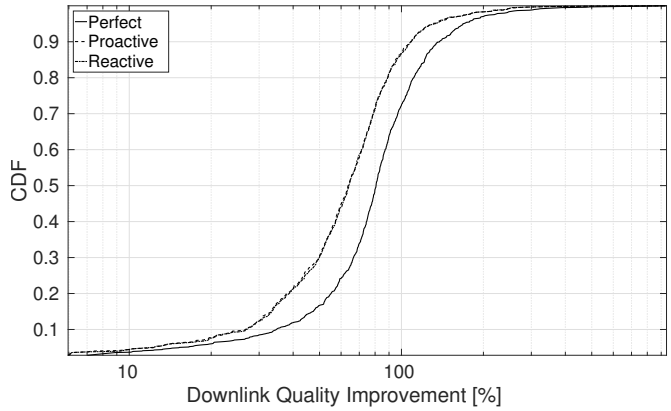


Fig. 6. CDFs of the data rate increase obtained by anticipatory networking solutions for different prediction accuracies.

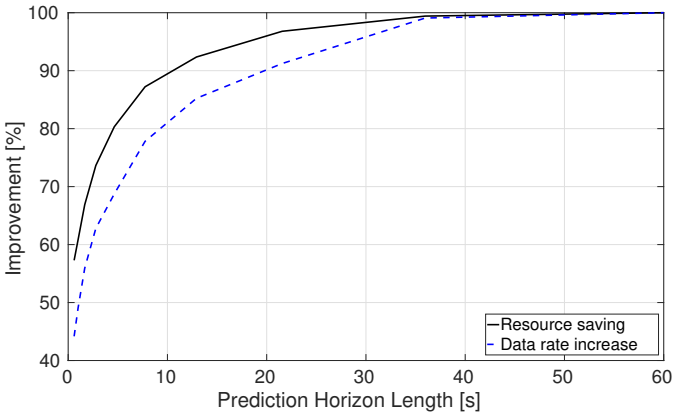


Fig. 7. Impact of the prediction horizon length.

barely reach 40%. This disparity of performance is justified by the different MCS statistics of the downlink and uplink channels, of which the second is consistently higher. In turn, this translate into a smaller margin of optimization for the uplink data rates, see Figure 2(c) for a comparison of the MCS CDF and Figure 3 for a detailed representation of MCS traces of the downlink channel. Overall, we measured data rate improvements between 20% and 100% with a median value of 65% for downlink communications and between 3% and 13% (median 6.5%) for the uplink.

We also compared the results of each location separately: the performance of our optimization framework does not differ by more than 5-10%, but for the Leganes dataset, which is due to the low load of this dataset. This means that anticipatory gains should be achievable regardless of the particular location.

To conclude this evaluation, we show in Figure 7 the impact of the prediction horizon. Basically, the prediction horizon represent the number of time slots optimized at the same time. Thus, a shorter horizon makes the optimizer less effective as it can only rely of short term information. In the figure we show normalized average results in order to be able to compare solutions with different performance. The chosen examples consider a maximum prediction horizon of one minute and analyze the same by giving the optimizer a fraction of the whole available information. Although the best performance is reached asymptotically, substantial improvements can be obtained with just a few seconds of prediction.

This last graph helps understanding why the realistic predictors performs so closely. In fact, reducing the prediction horizon of the omniscient predictor makes it similar to a realistic one which is more effective in the first time slots only. As such we can compare a realistic (either proactive or reactive) predictor to an omniscient one with an horizon of about 10 seconds.

A few final considerations about the overall approach are in order. The first concerns our datasets: optimizing network resource allocation starting from real traces makes it impossible for the optimizer to run into infeasible conditions, because the starting point was already feasible. The second consideration concerns whether the anticipatory gains can be estimated from the trace characteristics without solving the optimization problem. Studying the correlation between our final results and the compound metrics computed above for active users we found they are almost independent. This is due to the fact that the degree of improvement does not depend on the characteristics of individual users, but on their combination. Determining whether combining different users results in a good mix and provides high gains is a problem just as complex as the resource allocation problem itself.

V. Related works

In this section we discuss a few alternative approaches to our evaluation framework, alternative tools to record mobile network traffic and measurement-driven analysis of mobile networks. Yin et al. [19], [20] propose a throughput prediction solution based on clustering and hidden Markov models. Their predictor is subsequently used to control video bitrate selection in a multimedia streaming application. Finally, they evaluate their approach on a proprietary large dataset provided by a Chinese commercial video provider. Muppirisetty et al. [21] investigate the spatial prediction of wireless channels using Gaussian processes. Atawia et al. [22] focus on energy savings obtained thanks to predictive resource allocation and uncertainty management. Finally, Du at al. [23] design a predictive backpressure algorithm to solve the resource allocation problem for multimedia streaming.

These are just a few of the many papers adopting anticipatory networking and we encourage the interested reader to read further on the topic [7], where we provide a thorough review of the state of the art. The framework described here is not meant to provide yet another variation on the topic, but allowed us to test the performance of many realistic approaches against theoretical bounds on a big dataset.

If mobile operators disclose their datasets, very interesting and insightful papers originate. For instance, the recent works of Furno et al. [24], [25] study the influence of human activities on mobile communications and identify several traffic patterns that can be used to enhance anticipatory networking. In a similar fashion, Wang et al. [26] analyze the traffic in

Shangai and conclude that there are five main traffic profiles that represent most of the activity in the 9000+ studied cells. The same dataset is also analyzed by Ding et al [27] to model the network capability.

Differently from all these studies, we built our dataset using our LTE sniffer [15] and we plan to make our dataset available to the community to allow for comparative studies and the development of practical solutions. Our dataset, which is intrinsically anonymous due to the use of temporary identifiers instead of unique user IDs, is also the only one to provide scheduling information at millisecond granularity. Thus, to the best of our knowledge, our dataset is the only archive of mobile network traffic obtained independently of mobile operators.

VI. Conclusions

In this paper we conducted a performance evaluation of anticipatory networking solutions based on real-world traffic measurements. In particular, we provided three main contributions: a large dataset providing fine-grained scheduling information of four cells around Madrid, a comprehensive framework to study realistic anticipatory networking solutions against their theoretical bounds and a thorough evaluation of these techniques on our dataset.

In particular we found that anticipatory optimization provides substantial resource savings and data rate enhancements. In the dataset, more than 80% of the total traffic is produced by less than 30% of the users, which also exhibit the more predictable behaviors. Thus, by analyzing this predictable traffic component and considering the rest as inelastic background traffic, we obtained about 35% resource saving and 65% higher data rate in the downlink channel, while in the uplink channel lower gains are obtained due to the higher average MCS that is usually assigned in these communications.

We found that moving from an omniscient predictor to more realistic ones has a substantial impact on performance, while both proactive and reactive predictors achieved very similar performance. Finally, we conclude that anticipatory networking is both a viable and effective solution that merits is place in 5G networks. On top of the performance improvement, it provides a new perspective about dealing with context information that the network can provide to mobile operators and application developers to enable the services of the future.

Acknowledgment

This work has been supported by the European Union H2020-ICT grant 644399 (MONROE), by the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919), the Ramon y Cajal grant from the Spanish Ministry of Economy and Competitiveness RYC-2012-10788 and grant TEC2014-55713-R.

References

[1] J. L. Heilbron, *The dilemmas of an upright man: Max Planck and the fortunes of German science*. Harvard University Press, 1986.

[2] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feruary 2010.

[3] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *IEEE INFOCOM*, April 2011, pp. 882–890.

[4] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the limit of predictability in human mobility," *Nature Scientific reports*, vol. 3, October 2013.

[5] N. Bui, F. Michelinakis, and J. Widmer, "A Model for Throughput Prediction for Mobile Users," in *European Wireless*, May 2014, pp. 1–6.

[6] N. Sadek and A. Khotanzad, "Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model," in *IEEE ICC*, vol. 4, June 2004, pp. 2148–2152.

[7] N. Bui, M. Cesana, A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–1, April 2017.

[8] N. Bui, I. Malanchini, and J. Widmer, "Anticipatory admission control and resource allocation for media streaming in mobile networks," in *ACM MSWiM*, November 2015.

[9] N. Bui and J. Widmer, "Mobile network resource optimization under imperfect prediction," in *IEEE WoWMoM*, June 2015, pp. 1–9.

[10] N. Bui, S. Valentin, and J. Widmer, "Anticipatory quality-resource allocation for multi-user mobile video streaming," in *IEEE CNTCV (INFOCOM Workshop)*, April 2015.

[11] ETSI, "E-UTRA; Physical channel and modulation," *3GPP TS*, vol. 36.211, p. V13, 2016.

[12] N. Bui, F. Michelinakis, and J. Widmer, "Fine-grained LTE Radio Link Estimation for Mobile Phones," in *IEEE WoWMoM*, June 2017.

[13] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *ACM MobiCom*, October 2016.

[14] S. Makridakis and M. Hibon, "ARMA Models and the Box–Jenkins Methodology," *Journal of Forecasting*, vol. 16, no. 3, pp. 147–163, 1997.

[15] N. Bui and J. Widmer, "OWL: a Reliable Online Watcher for LTE Control Channel Measurements," in *ACM ATC (MobiCom workshop)*, October 2016.

[16] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An Open-Source Platform for LTE Evolution and Experimentation," in *ACM ATC (MobiCom workshop)*, October 2016.

[17] Cisco VNI, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," *Cisco Public Information*, 2016.

[18] IBM, "ILOG CPLEX," https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, last accessed June 2017.

[19] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 325–338, August 2015.

[20] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *ACM SIGCOMM*, August 2016, pp. 272–285.

[21] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1031–1044, May 2016.

[22] R. Atawia, H. Abou-zeid, H. S. Hassanein, and A. Noureldin, "Joint chance-constrained predictive resource allocation for energy-efficient video streaming," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 34, no. 5, pp. 1389–1404, May 2016.

[23] J. Du, C. Jiang, Y. Qian, Z. Han, and Y. Ren, "Resource allocation with video traffic prediction in cloud-based space systems," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 820–830, May 2016.

[24] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, "A tale of ten cities: Characterizing signatures of mobile traffic in urban areas," *IEEE Transactions on Mobile Computing*, December 2016.

[25] A. Furno, M. Fiore, and R. Stanica, "Joint spatial and temporal classification of mobile traffic demands," in *IEEE INFOCOM*, April 2017.

[26] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," in *ACM IMC*, October 2015, pp. 225–238.

[27] J. Ding, X. Liu, Y. Li, D. Wu, D. Jin, and S. Chen, "Measurement-driven capability modeling for mobile network in large-scale urban environment," in *IEEE MASS*, January 2016, pp. 92–100.