Predicting file downloading time in cellular network: Large-Scale analysis of machine learning approaches

Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, Gwendal Simon

Predicting File Downloading Time in Cellular Network: Large-Scale Analysis of Machine Learning Approaches

Alassane Samba[a,*], Yann Busnel[b], Alberto Blanc[b], Philippe Dooze[a], Gwendal Simon[b]

[a] *Orange Labs, Lannion, France*
[b] *IMT Atlantique / IRISA / UBL, Rennes, France*

Abstract

Downlink data rates can vary significantly in cellular networks, with a potentially non-negligible effect on the user experience. Content providers address this problem by using different representations (*e.g.*, picture resolution, video resolution and rate) of the same content and switch among these based on measurements collected during the connection. Knowing the achievable data rate before the connection establishment should definitely help content providers to choose the most appropriate representation from the very beginning. We have conducted several large measurement campaigns involving a panel of users connected to a production network in France, to determine whether it is possible to predict the achievable data rate using measurements collected, before establishing the connection to the content provider, on the operator's network and on the mobile node. We establish evidence that it is indeed possible to exploit these measurements to predict, with an acceptable accuracy, the achievable data rate. We thus introduce cooperation strategies between the mobile user, the network operator and the content provider to implement such anticipatory solution.

Keywords: Throughput prediction, Cellular networks, Machine learning,

*Corresponding author
Email addresses: alassane.samba@orange.com (Alassane Samba), yann.busnel@imt-atlantique.fr (Yann Busnel), alberto.blanc@imt-atlantique.fr (Alberto Blanc), philippe.dooze@orange.com (Philippe Dooze), gwendal.simon@imt-atlantique.fr (Gwendal Simon)

1. Introduction

In cellular networks, Quality of Service (QoS), in particular the throughput, can significantly vary depending on factors like the wireless channel conditions, users' activity, and location. To deal with changing QoS, content providers implement *adaptive* delivery strategies, where the quality and the characteristics of the delivered content are adjusted to match the achievable QoS of each user. These adaptive strategies are reactive: the characteristics of the content delivered at time t are based on measurements collected between the beginning of the connection and t.

Yet, content providers should take some key decisions at the beginning of the delivery. For instance, most web sites have different versions of their pages, with a variable number of elements and different versions, e.g., pictures with different resolutions [1, 2]. The decision of which version to deliver must be taken at the very beginning of the connection, even though no past observation of the network path performance is available. Another example is adaptive video streaming, where video is divided into several chunks, and each chunk encoded at different bit-rate, each one corresponding to a quality level. Throughout the connection, the client selects the highest quality representation with respect to an estimation of the available bandwidth based on the most recent history. At the very beginning, though, no such history is available. The delivery often starts with a medium or low quality representation to be on the safe side [3]. In the case of interactive communications like voice and video calls, the service provider (or the application) must select the data rate used to encode the voice and video signals at the beginning, and then this rate changes slowly [4]. Similarly, the settings of cloud-based applications with stringent real-time constraints, such as cloud gaming [5], should be done at the very beginning of the session and can sometimes hardly be changed (for instance the choice of the server). In each of these cases, content providers could avoid the guess work if

they could get a reasonably accurate estimate of the achievable data rate of a given client. This estimate does not need to be extremely precise. Getting the order of magnitude right can already be enough in most cases. Typically, the number of web page versions (for a web provider) and the number of available video representations (for a video provider) are limited.

The abundant literature on throughput prediction, which is detailed in Section 2, does not meet our demand for *historyless* estimation of the time needed to download a file. Many previous proposals for throughput prediction are based on end-to-end (or path) metrics, such as the Round-Trip Time (RTT) and the packet loss rate. While these proposals can be very effective, they need measurements of end-to-end metrics, which are not available at the very beginning of the connection. A second group, sometime called history-based, relies on the throughput history, as its name implies. These techniques, which are mainly used in adaptive video streaming, predict the throughput in the immediate future [6, 7, 8]. As in the case of proposals based on end-to-end measurements, history-based proposals are not an option at the very beginning of a connection. More recently, Lu et al. [9] have proposed to exploit physical layer measurements to make very short term (a few hundreds of milliseconds at most) throughput predictions. While such short term predictions can be useful for congestion control and packet scheduling purposes, they are less relevant for content providers as the time scales involved are too short: clients need a few seconds to download the typical web page or video segment.

Our goal in this paper is to study whether it is possible to get an accurate estimate of the time needed to download the first few elements requested by the client (for instance the different components of a web page or the first segments of a video). In other words, we study the prediction of the *average throughput* on a time period that is much longer than what was studied in the aforementioned recent proposals. We focus on the case of cellular networks, which are a challenging environment since the QoS performance can change dramatically.

The key idea of our prediction approach is to use machine learning algorithms

on a set of data coming from different sources, such as radio channel metrics (e.g., received signal strength), Global Positioning System (GPS) coordinates, speed, terminal category, frequency band of the mobile, and cellular network metrics (e.g., average cell throughput, the average number of users, the connection success rate). These different elements are available either on the mobile station and/or cellular base station and in the cellular network management system. We have presented some preliminary results in a previous paper [10] in which we verified on a small dataset the validity of our approach.

In this paper, we comprehensively study the idea of using machine learning to predict, without any prior end-to-end measurement, the time needed to download a file. For the sake of completeness, we use a large dataset, which we collected at different time periods in a production mobile network, and four machine learning algorithms. Our main contributions are the following: (i) we analyze the bivariate relationship between each input variable and the throughput, to better understand the contribution of each variable to the cellular network throughput prediction; (ii) we show that an instantaneous prediction of user achievable throughput is possible; (iii) we highlight that the accuracy depends on the type of information in input; (iv) we discuss possible implementations of this approach to improve the adaptive delivery strategies.

The remainder of the paper is organized as follows: After a review of related work in Section 2, we describe the measurement campaigns and the input dataset in Section 3. We analyze the bivariate correlation between different input variables and the throughput in Section 4. We then exploit these results in the comparison of different machine learning algorithms to predict the throughput in Section 5. Finally, we discuss our approach and introduce implementation strategies in Section 6.

2. Related work

We review, in the following, previous work on throughput prediction. We distinguish three main families of studies, whether the prediction is based on

knowledge of the end-to-end path characteristics, on measured throughput history, and on information about the physical layer. We conclude by summarizing the main differences between our work and existing solutions.

2.1. End-to-End Path Knowledge-Based Throughput Prediction

Scientists have analyzed the behavior of Transmission Control Protocol (TCP) to obtain accurate models for the prediction of the instantaneous throughput on a given end-to-end path. Various models have been studied for the past twenty years [11, 12, 13, 14, 15, 16, 17]. The theoretical findings include the strong ties between the throughput dynamics, the RTT, and the packet loss rate. It is possible to get an accurate estimation of any of these end-to-end parameters from the two others. These studies however do not meet our motivations to estimate the throughput before the beginning of the connection, when no past measurement is available. Since we consider that the system is aware of neither the RTT nor the packet loss rate, we cannot directly use these models to obtain an estimate of the average throughput on a typical 10 s period.

Another set of scientific studies [18, 19] has aimed at replacing the end-to-end parameters in the throughput models by a set of other metrics that can be obtained quickly by light path probing. The prediction error is however less accurate than with RTT and packet loss rate due to the uncertainty on the validity of the probed metrics. Moreover our motivation is to have no probing phase at all, and to rely exclusively on information that can be instantaneously available before the beginning of the connection.

2.2. History-Based Throughput Prediction

A line of research on throughput prediction is the development of models that are based on past measures of the throughput. The motivation is to use approaches related to time series analysis (e.g., smoothing or moving average) to forecast throughput by observing its past evolution. For example, Xu et al. [20] provide a framework dedicated to network performance forecasting, including forecasting the throughput during the next 500 ms based on the last 20 s.

In a similar vein, Jiang et al. [21] use a harmonic mean of the previously observed throughput. Li et al. [22] also use a smoothing function to predict the throughput. Sun et al. [23] divide the session into periods of 6 s (*epochs*) and use a Hidden Markov Model (HMM) to predict the throughput of each period.

Adaptive video streaming technologies use a variant of the history-based throughput prediction to determine which bit-rate is the most appropriate for the next x seconds (x ranging from 1 to 10 s) [21, 22]. The prediction is based on buffer monitoring techniques [6], where the buffer level drives the video rate selection. However, when the buffer is empty at the beginning of the streaming session, a bandwidth estimation is necessary, as highlighted by Mok et al. [3].

While having proven effective in wired networks [20, 21, 22], these strategies do not perform as well in cellular networks due to the frequent radio link quality variations, which influence the bandwidth. Yao et al. [24] have highlighted this weakness showing that, in cellular networks, the current throughput should rather be predicted by the user location, which has a stronger impact on radio link quality. More recently, Zou et al. [8] study more accurate bandwidth prediction methods with respect to radio link quality variations.

These approaches do not address our motivation for getting an estimation (which does not need to be as accurate as in these proposals) without any previous history available at the client side. We study in a comprehensive manner the combination of various available information, in particular related to user location, to identify the set of parameters that have the highest influence of throughput prediction.

2.3. Physical Layer-Based Throughput Prediction

A more recent approach is to predict throughput by using measurements that have been collected at the physical layer by the network operator. Our paper falls into this category, which has been recently explored for cellular networks in a series of papers [9, 25].

Lu et al. [9] rely on the fact that the mobile nodes and the base stations periodically exchange radio channel measurements, which are used by the base

station to make scheduling decisions. Among these measurements, Lu et al. suggest to use the Channel Quality Indicator (CQI), which they directly map to a throughput estimation. The suggested estimation is valid only for a short time frame (500 ms is the longest tested time frame [9]), as the CQI is very dynamic. This approach does thus not fit our needs since our motivation is to address the downloading of data chunks that can be several seconds long (up to 15 s for a video segment [26]). More generally, we are interested in predicting the time needed to download a large file.

The framework presented by Xie et al. [25] aims to infer the available bandwidth from the Physical Resource Blocks (PRBs) utilization rate. Their proposal is to estimate a "sojourn" time of the network bandwidth (the validity duration of this bandwidth estimation) using a Pareto model. The approach of estimating the PRBs utilization is equivalent to exploiting directly the Reference Signal Received Quality (RSRQ), which already provides the channel quality information without the need of implementing another radio resource monitoring system. This approach does not enable an instantaneous prediction, as the previous bandwidth value and its sojourn time must be constantly recorded.

The main difference between these approaches and ours is that we collect data from both user devices and base stations, including physical layer and aggregated cell performance data. Our goal is to extend this series of work by having a systematic analysis of all the available measurements (without probing the end-to-end path) and their significance as throughput predictors.

2.4. Summary

The popularity of adaptive delivery technologies has reinforced the need of a throughout prediction that is both reasonably accurate and that does not rely on time consuming path probing. We reviewed the existing approaches, which can be based on end-to-end path performance analysis, on the history of throughput measurements, and on some physical layer information. Only the latter approach provides an instantaneous prediction and is well suited for cellular

networks. Previous studies are limited to short time scales (e.g., 500 ms in [9]). In this paper, we complete the physical layer-based instantaneous throughput prediction by adding as input several aggregated Radio Access Network (RAN) performance data and user context information to obtain a more accurate prediction of the average throughputfor a longer time scale. We also compare the inputs by measuring the contribution of each one to the prediction.

3. Input Dataset

To evaluate our approach on real datasets, we conducted a series of measurement campaigns in 2016, involving more than 50 volunteers spread over several cities in France (Europe). In the following, we first give an overview of the different campaigns. Then, we present in Section 3.2 a detailed description of the collected data. Finally, the preprocessing of the data (including cleaning and filtering) is detailed in Section 3.3.

3.1. Measurement Campaign

Each User Equipment (UE) periodically downloaded a specific file through an Android application [27] from a dedicated well-provisioned server. The size of the downloaded file ranges from 4 to 50 MB, depending on the measurement campaign. This range of sizes allows us to test our prediction solutions in a wide set of configurations. To make a few comparison, the average size of a webpage has exploded in the recent years and is planned to be above 4 MB in 2019 [28]. Based on the recommended video encoder setting, a typical 2 s chunk of a 4K video is above 10 MB [29]. The time needed to download these files were in the order of a few seconds, which is in conformance of our motivation.

The measurements were performed on a production Long Term Evolution (LTE) network, using TCP. For each measurement, the application logged various data from the UE Operating System (OS), at the beginning and at the end of the connection. Meanwhile, a series of measurements regarding the cellular network performance was collected in parallel from the operator RAN management system, which include dozens of Key Performance Indicators (KPIs). An

Table 1: Main characteristics of the dataset

	campaign			
	A	**B**	**C**	**all**
General Information				
dates	early 2016	early 2016	end of 2016	
nb. testers	40	5	5	
nb. entries	5,441	2,632	15,011	23,084
nb. phone models	23	5	5	26
nb. cells	390	40	122	497
nb. cities	112	14	41	147
File Size				
4MB / 20MB / 50MB	5,441 / 0 / 0	0 / 2,632 / 0	0 / 12,437 / 2,574	5,441 / 15,069 / 2,574
Cell Frequency Band				
proportion of LTE2600 / LTE800	0.58 / 0.42	0.88 / 0.12	0.63 / 0.37	0.64 / 0.36
Device Category				
proportion of cat.3	0.23	0.00	0.00	0.05
proportion of cat.4	0.70	0.59	0.77	0.74
proportion of cat.6	0.07	0.41	0.23	0.21
Context				
proportion of indoor / outdoor / unknown	0.41 / 0.15 / 0.44	0.47 / 0.31 / 0.22	0.59 / 0.26 / 0.15	0.52 / 0.24 / 0.23
proportion of static / mobile / unknown	0.57 / 0.08 / 0.35	0.60 / 0.20 / 0.20	0.83 / 0.02 / 0.15	0.74 / 0.06 / 0.20
median / average speed (in m/s)	0.00 / 0.45	0.00 / 0.39	0.00 / 0.14	0.00 / 0.23
median / average distance to base station (in m)	857 / 1,302	702 / 1,387	2,931 / 2,777	1,879 / 2,336
Radio				
median / average RSRQ (in dB)	-7.7 / -8.3	-6 / -7	-8 / -8	-8 / -8
median / average RSRP (in dBm)	-106 / -104	-110 / -109	-111 / -109	-111 / -108
median / average RSSI (in dBm)	-79 / -80	-83 / -83	-80 / -82	-80 / -82
RAN				
median / average cell avg. throughput (in Mbit/s)	29 / 27	30 / 29	29 / 28	29 / 28
median / average avg. nb. of user	14 / 20	2 / 9	9 / 16	7 / 15
median / average block error rate	0.05 / 0.06	0.05 / 0.05	0.08 / 0.09	0.07 / 0.07
median / average conn. success rate	0.999 / 0.997	1.000 / 0.999	0.999 / 0.998	0.999 / 0.998
Throughput (in Mbit/s)				
1st quartile / median / 3rd quartile	13 / 23 / 36	25 / 37 / 46	11 / 24 / 34	12 / 26 / 36
average	25	34	23	25

exhaustive description of the collected data is provided in Section 3.2. Figure 1 illustrates the different elements involved.

In more details, we undertook three measurement campaigns (respectively identified in the following by A, B and C), differing in context, downloaded file sizes, and mobility patterns of the UEs. This diversity is representative of the typical settings that are observed in the network. During each campaign, the application has configured to periodically perform measurements throughout the whole day for all the UEs involved in the campaign. Table 1 presents the details of each campaign.

Campaign A was held early 2016. It involved 40 volunteers spread over several cities in France. This campaign presents a large diversity of cells, with an

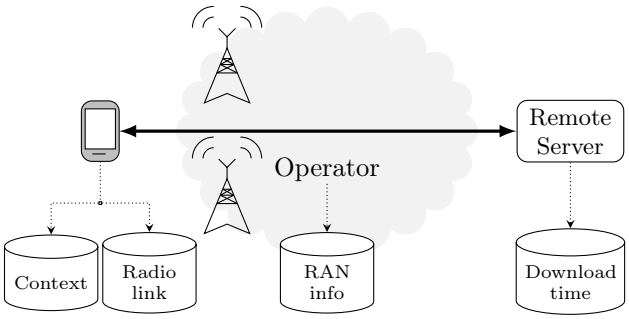

Figure 1: Overall architecture of our data collection campaign

emphasis on urban contexts. To avoid overusing the data allowance of the volunteers' wireless plans, we limited the download size to 4 MB.

Campaign B was held during the same period as Campaign A, but involved five test smartphones with dedicated SIM cards associated to unlimited data plans. The download size was then fixed to 20 MB for these UEs. This campaign was held in better cellular conditions by users in well-covered mid-size and touristic cities, but it presents a lower diversity of cells.

Campaign C was held during end of 2016, using the same five test smartphones as in Campaign B. It has been performed to collect data over more various conditions. In this campaign, the download size was set to 50 MB for one sixth of the measurements, and 20 MB otherwise. It represents mostly static usage in various locations (for instance, connection from different houses), mostly in rural areas.

The collected data consist of 23,084 entries. This includes only the smartphones that successfully performed the tests. 26 different smartphones models have been used, with different LTE UE categories [30] (6 in *cat. 3*, 18 in *cat. 4* and 2 in *cat. 6*). About 500 cells, which are spread over 147 communes in France, appear in the data, which are split into 64% LTE2600 and 36% LTE800. In order to avoid measurement artifacts, connections that experienced a handover are not considered in this study.

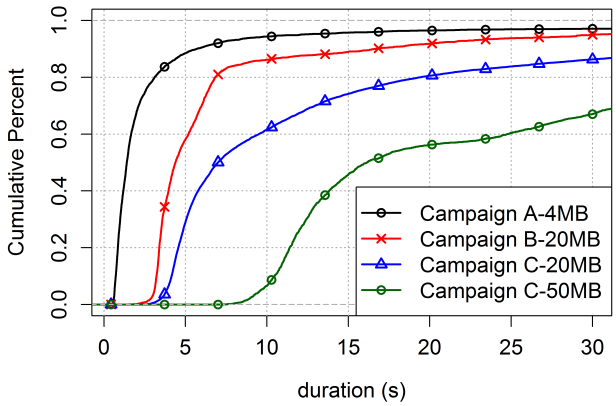

Figure 2: Cumulative distribution function of the file download time

3.2. Dataset Description

We now describe the metrics collected in the measurement campaigns, grouped by type from the business and accessibility points of view.

3.2.1. Average Throughput

The main measure that we target is referred to as the *average throughput*, which we compute as the file size divided by the downloading time. TCP [31] includes a congestion control algorithm that is designed to adjust the sending rate to the available bandwidth. A TCP session includes two phases: slow-start and congestion avoidance. The former corresponds to a rapid increase of the sending rate from the beginning until a threshold is reached or packet losses are detected. Then, the congestion avoidance phase starts, where the sender increases the sending rate until a packet loss is detected and the rate is reduced. Different TCP versions use different congestion control algorithms to increase the rate and use different multiplicative factors when reducing it but the main idea is the same, at least for congestion control based on loss detection, which are the most widely used.

We used different file sizes during our measurement campaign to cover various cases: small files to measure the average throughput in the case of networks with low bandwidth without extending the transmission duration. Large files to

measure network configuration with high bandwidth in not too short sessions. Some of our measurements in campaign A-4 MB may have stopped before the end of the TCP slow start. Nevertheless, since 70 % of the sessions lasted more than 1 s for this campaign, we believe that the range of its session durations is large enough to be representative of typical network configurations. The empirical cumulative distribution function is shown in Figure 2. For the campaign B-20 MB (resp. C-20 MB and C-50 MB), all the sessions lasted more than 1.5 s (resp. 2.5 s and 7 s).

3.2.2. UE Category and Cell Frequency Band

LTE defines UE categories, which determine their performance specifications and enable base stations to be aware of their expected performance level.

Cellular networks use several frequency bands: 2.6 GHz for LTE2600 and 800 MHz for LTE800 base stations. LTE2600 is particularly used in city-based scenarios since it uses a larger band (20 MHz) and does not cover a wide area. LTE800 uses a smaller band (10 MHz). Its carrier frequency is generally used in a rural setting to take advantage of its wider coverage.

3.2.3. Physical Layer

On the UE we collect:

Received Signal Strength Indicator (RSSI) the total wide-band signal power received by the UE.

Reference Signal Received Power (RSRP) the reference signal power across the channel bandwidth.

Reference Signal Received Quality (RSRQ) the ratio between RSRP and Received Signal Strength Indicator (RSSI) multiplied by the number of resources blocks allocated to the UE. RSRQ measures the quality of reception of the reference signal.

While other metrics such as Signal to Interference and Noise Ratio (SINR) and CQI do exist, we have observed that most UEs do not report accurate values,

possibly because of the OS and the device itself, or most likely, a combination of both.

The Application Programming Interface (API) offered by the OS, allows applications to obtain information about the UE. On Android UEs radio data availability depends on the OS version and the energy saving policies of the manufacturer. For example, we discovered that a screen in *idle* mode can shut down the access to some radio link KPIson some UEs. Since our measurements have been performed on each smartphone without specific user intervention, many tests have been done while the smartphone was idle, so many logs do not contain all the data.

It is worth noting that CQI data are no longer available in more recent Android UEs. This explains why Lu et al. [9] use a complex setup using Qualcomm's QXDM software to capture radio layer traces including CQI reports. Since we focus on a lightweight data collection and throughput prediction, we ignore the CQI in our campaign and used RSRQ as our main quality indicator.

3.2.4. Radio Access Network

Operators use Network Manangement System (NMS) to monitor their networks by collecting raw counters of network events, typically aggregated over a period of a few minutes. Among all the available metrics, we selected those that are both well correlated to the user throughput and least correlated between them. They are: the average cell throughput, the average number of users in the cell, the Block Error Ratio (BLER) of the cell, and the Radio Resource Control (RRC) setup success rate. Each metric is computed over a period of fifteen minutes.

Finally, we would like to note that, despite the relative stability of the NMS, we also missed a part of RAN data due to some data gathering issues.

3.2.5. Context Information

Intuitively, the context in which the download operation occurs can help to predict QoS. Context awareness has been used in various other applications [32,

Table 2: Definition of the ten main schemas for the entries

				radio			context			RAN				
	Schema	UE cat.	cell freq. band	RSRQ	RSRP	RSSI	distance	speed	in/outdoor	avg. throughput	avg. nb. of user	BLER	conn. success rate	nb. occurrences
	s_1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	9,459
	s_2	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	3,347
	s_3	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	1,445
	s_4	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	1,232
	s_5	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	1,224
	s_6	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	1,079
	s_7	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	731
	s_8	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	572
	s_9	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	540
	s_{10}	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	502
Total missing		0	0	3,587	4,877	9,351	4,638	4,679	5,234	7,543	7,543	5,357	5,357	

33]. In this paper, we consider the following indicators:

Indoor/Outdoor: provided by the application that we have used on the UEs (heuristic based on the number of visible GPS satellites).

Distance to cell: based on the GPS coordinates provided by the application and the topology database with the location of the base stations of the cellular network.

Speed: estimated thanks to the GPS and the accelerometer.

3.3. Data Preprocessing

Every time a UE downloads the file, it generates an *entry*. This entry contains a timestamp and the time it took to download the full file, which we use to compute what we call average throughput. An entry also contains the measurements that we presented in the previous section.

Some entries are incomplete. For example, only a subset of the radio channel measurements is available, depending on the make and model of the UE as well as the version of the OS. The location (GPS coordinates) is not always available too. Finally, some devices do not report radio link measurements when in idle mode. We have thus to deal with some missing data.

Table 3: Correlation between collected information and throughput

			Radio			**Context**			**RAN**			
	UE cat.	cell freq. band	RSRQ	RSRP	RSSI	distance	speed	location	cell avg. throughput	cell avg. nb. of user	cell BLER	cell conn. success rate
Pearson correlation coef.	-	-	0.67	0.39	-0.17	-0.64	0.00	-	0.59	-0.53	-0.39	0.07
Spearman correlation coef.	-	-	0.71	0.45	-0.15	-0.66	0.11	-	0.62	-0.57	-0.43	0.47
MIC	-	-	0.37	0.18	0.33	0.62	0.07	-	0.49	0.30	0.20	0.24
Point-biserial correlation coef.	-	0.69	-	-	-	-	-	0.09	-	-	-	-
MaxNMI	0.09	0.48	0.32	0.14	0.32	0.49	0.02	0.03	0.38	0.24	0.13	0.19

We distinguish ten different cases, depending on the missing metrics. We call them *schemas*. Each schema is characterized by the missing metrics: all the entries belonging to the schema have the same missing metrics and have valid measurements for all the others. Table 2 shows the ten most frequent schemas in our measurement campaign. The schema s_1, in which all metrics are present, is the most frequent with 9,459 entries. We were able to exploit the other schemas as well given that the prediction algorithm (presented in Section 5) can use only a subset of the available metrics. For instance, when using the radio information and the context information, we include schemas s_1 and s_2.

We preferred this approach of mixing different schemas. The other traditional approaches to deal with missing data are either *imputation* — the missing values are replaced by the mean of the variable, its median or any other value learned from the dataset [34] —, or *entry removal* — the whole entry is ignored. Both approaches do not fit well with our problem. In the case of imputation, the choice of the imputed value can have a significant impact on the results and there is no value which is obviously appropriate for such a purpose. The approach based on entry removal would significantly reduce the amount of entries for our supervised learning approaches.

4. Bi-Variate Correlations

We first calculate the several *correlation coefficients* to study the bi-variate correlations. We use *linkspotter* [35], a package for the *R statistical software* [36]. We provide a summary of the correlations in Table 3.

Pearson coefficient indicates a linear positive (or negative) relationship between the variable and the throughput. The closer the value to is to 1 (respectively -1), the higher the correlation is (inverse correlation respectively).

Spearman coefficient is comparable to the *Pearson* coefficient, but rather denotes a monotonic relationship between the variable and the throughput.

Maximal Information Coefficient (MIC) **[37]** indicates several types of correlation including linear, non linear, monotonic and non monotonic ones. The closer it is to 1, the higher the correlation.

Point-biserial coefficient is similar to the Pearson coefficient but applies to binary variables.

MaxNMI **[35] correlation coefficient** (Maximal Equal-Freq Discretization-Based Normalized Mutual Information) enables the evaluation of the correlation between any couple of variables regardless of their types (qualitative and quantitative). Any quantitative variable is discretized into intervals of equal frequency so as to maximize the normalized mutual information of the couple. With this normalization, we obtain a coefficient between 0 and 1 (the closer it is to 1, the stronger the relationship is), which can be compared between different couples of variables. Moreover, the discretization fulfills a condition to avoid overfitting data and to limit the calculation time [37].

Based on the results given in Table 3, we identify the metrics that are loosely correlated to throughput: the user location (in or out-door), the speed, and the device category. The two former measurements are hard to obtain accurately, so it is possible that the low correlationis due to a failure in the data collection process.

We provide a general graphical view of the correlation between each collected metric and the throughput by a series of figures: for the cell frequency band,

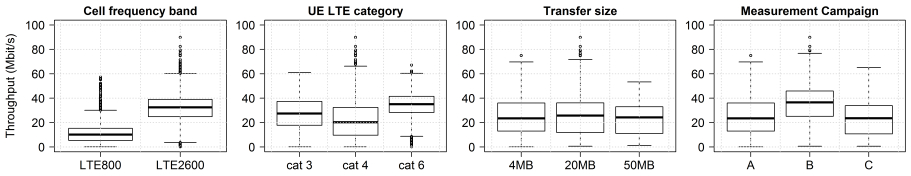

Figure 3: Throughput according to other factors

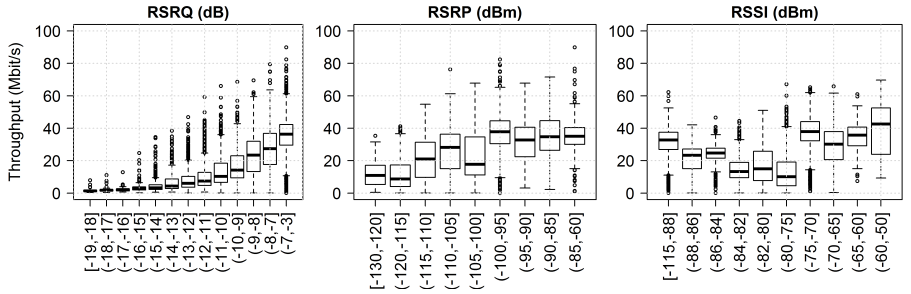

Figure 4: Correlation between throughput and radio data

the UE category, the transfer size and measurement campaignin Figure 3, for the radio link datain Figure 4, for the context datain Figure 5, and for the RAN datain Figure 6. We present the results using *box plots*. They describe the entire distribution of the average throughput on each interval obtained by an equal-frequency discretization of the corresponding KPI. A box represents the quartiles and its whiskers the range of the data (excluding potential outliers).

Our observations include: (i) the high correlation between the cell frequency band and the throughput is visible in Figure 3 where the connection to a LTE2600 has a better throughput than a connection to a LTE800 cell. The higher throughput of LTE2600 may be due to the larger bandwidth, which allows higher maximal data rate. (ii) Figure 3 illustrates the low correlation between the throughput of the transfer size and the campaign. The differences between the three campaigns, particularly the transfer size, have not impacted the throughput. Nevertheless, the campaign B obtained higher throughputs due to the over-representation in this campaign of LTE2600 cells and lightly

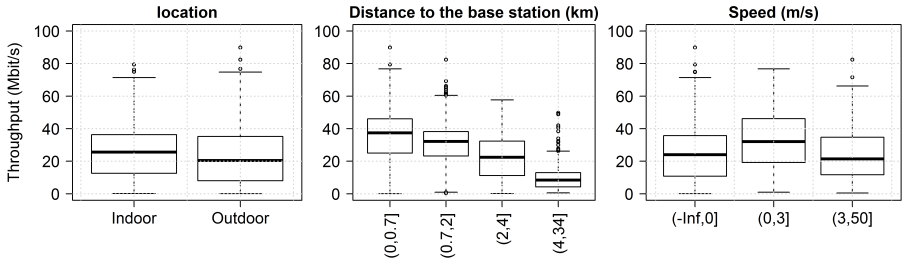

Figure 5: Correlation between throughput and context data

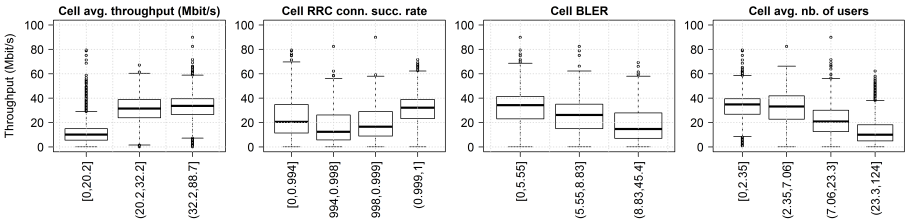

Figure 6: Correlation between throughput and RAN data

loaded cells (see Table 1). (iii) the radio link information exhibits a high correlation with throughput in Figure 4 but the number of outliers is still significant. (iv) Figure 5 denotes an inverse correlation between the distance to the base station and the throughput. The larger distance values from the base station correspond to the most frequent tests performed on LTE800 cells, which have a wider coverage and generally provide a lower throughput as explained in Section 3.2. (v) Figure 6 highlights that the four metrics that we chose from the RAN entries have a high correlation with the throughput. These metrics reflect several high impact factors related to the cell.

5. Input Combination for Prediction

In the previous section, we analyzed the bi-variate correlation between each metric and the throughput. Now, we expand our study by considering a throughput prediction based on a combination of these metrics. In our study, we check the accuracy of the throughput prediction based on a combination of metrics

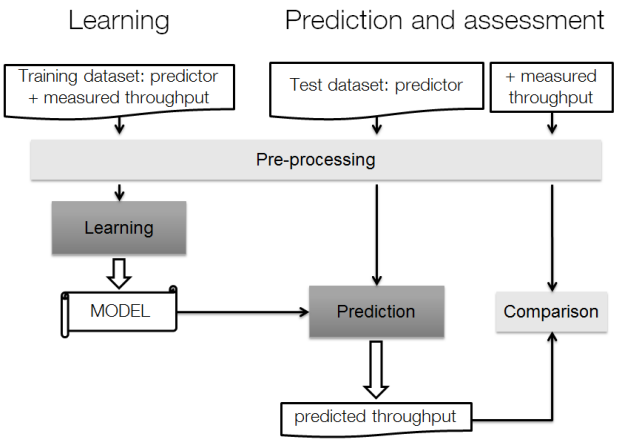

Figure 7: The machine learning process

as input. We call *predictor* every combination of input metrics that can be considered for the prediction. We consider predictors reflecting the data classification presented in Section 3.2. For each predictor we performed a machine learning process as described by Figure 7. The first phase consists in building a predictive model from a pre-processed training dataset, composed by the predictor metrics and the measured throughput, using a learning algorithm. In the second phase, we compute a predicted throughput from the predictor in a pre-processed test dataset with to the corresponding prediction algorithm and assess the prediction accuracy by a comparison with the measured throughput in a test dataset.

5.1. Methodology

The predictors are built as follows. First, we always include the UE category and the cell frequency band since both metrics are always available. Then, we consider the different types of metrics, as described in Section 3.2: context, radio link, and RAN. We consider all the configurations of available metrics for each type. Let i be an entry of our measurement campaign. Let y_i be the throughput of the file downloading operation associated with i. The subset of

metrics that has been accurately captured during the entry i corresponds to one of the schemas, which we introduced in Table 2. Based on the schema, the entry i can be used in some predictors. For example, an entry i that matches the schema s_1 can be used for three different predictors: when only RAN information is available, when only radio link information is available, and when both RAN and radio link information is available. For each predictor, the algorithm predicts the throughput $\hat{y_i}$. We then compare the predicted throughput $\hat{y_i}$ to the actual throughput y_i.

We use four *supervised learning* algorithms to make sure that the accuracy of the prediction from a given predictor is not due to a bias in the learning algorithm but rather in the correlation between this predictor and the throughput. Supervised learning is a field of machine learning to learn the relationship between a set of variables called *input* or *predictors* and another called *output* or *target*. We use the following algorithms: Random Forest (RF) [38], a Linear Model (LM), a particular design of a Generalized Linear Model (GLM) [39] using Partial Least Squares (PLS-GLM) [40], and an artificial Neural Network (NN) [41]. We describe each of these algorithms in Section 5.3. We use the *R statistical software* [36] and several packages that implement these algorithms: *randomForest* [42], *stats* [36], an extension that we coded from the package *pls* [43], and *nnet*.

To evaluate the algorithms, we randomlyseparate the dataset into two subsets: a training one (60%) and a test one (40%). The training dataset is used to build the models using the machine learning algorithms. The prediction is then done using input variables from the test dataset. This validationmethodology is a standard machinelearning technique to check whether a model that has been built with a (hopefully large enough) number of entries can then accurately predict the throughput of any new entry.

5.2. Performance Metrics

We measure the accuracy of the prediction from different perspectives, in order to avoid biases due to the performance metrics. In Table 4 presents two

metrics that are frequently used in machine learning:

The coefficient of determination, R^2 , represents the proportion of the variance in the throughput that is predictable from the predictor. It is computed as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y} - y_i)^2} \quad (1)$$

where $\bar{y}$ is the mean throughput calculated from the test dataset consisting of n measurements.

The median absolute error ratio, $\bar{E}$ is the error ratio that half of the predictions reach. It measures the median deviation of the prediction as a percentage. The error ratio is measured by the absolute value of the difference between the predicted and the actual throughput, divided by the actual throughput:

$$\bar{E} = \text{median}_{1 \leq i \leq n} \left(\left| \frac{y_i - \hat{y}_i}{y_i} \right| \right) \quad (2)$$

We also present the results in a graphical way by depicting the Empirical Cumulative Distribution Function (ECDF) of the prediction error in Figure 8 and the generated *heatmap* comparing actual measured and predicted throughput values in Figure 9. For both, we restrict to the three main predictors: radio-only, RAN-only, and radio and RAN.

5.3. Machine Learning Algorithms

We give now a brief overview of the four algorithms that we use to predict the throughput from the different predictors.

5.3.1. Random Forest Regression

RF regression algorithm is an extension of Regression Trees. The principle is as follows: first a set of regression trees is built using a randomly chosen number of predictor variables. Every tree is built using a *bootstrap* sampling [44]. The set of random regression trees is called the *random forest*. A prediction is

performed using the mean value of the output from all the trees in the forest. In this case, we use for each model a forest of 500 trees, and each tree is built using a randomly selected half of the predictor variables.

RF presents some appealing features. This method is known to be smoother than many other regression algorithms, in particular the Regression Tree. RF is sensitive to complex relationships between predictor variables and target data. Neither the outliers nor overfitting of the training set impact its prediction performance. Another advantage is that RF can be easily tuned since the number of learning parameters is low. Finally, the learning algorithm supports parallel execution.

5.3.2. Linear Model

The *LM* approach fits a linear equation $Y + \epsilon = \beta X$ to the data, where Y is the prediction, X the predictor variables matrix, β the coefficients to estimate, and ϵ the error. Given the assumption that the errors ϵ follow a Gaussian distribution and are *homoscedastic*, the approach is to use an optimization algorithm to estimate the coefficients β from a series of statistical tests. Then, the algorithm can compute a predicted value $\hat{Y}$ from any new predictor values.

The main advantage of LM approaches is the simplicity, which results in a small number of parameter settings and fast computation. However, LM can fail to detect some correlations between the target variable and the predictors. Moreover, the presence of outliers can affect the accuracy of the prediction.

5.3.3. Partial Least Square Generalized Linear Model (PLS-GLM)

The PLS-GLM technique [40] is a variation based on GLM, which aims in particular at fixing the inaccuracy of LM to detect correlations such as $\log(Y + \epsilon) = \beta X$. It also detects when some variables of the predictor are highly correlated; it is then possible to reduce the dimension of the predictor by applying a Multiple Correspondence Analysis (MCA) process to the input.

PLS-GLM is expected to improve LM while keeping its main advantages (including simple tuning and low execution time). Moreover, it takes into ac-

Table 4: Prediction results according to input data. B is the baseline predictor where the input variables are the UE category and the cell band.

		training	**test**	**RF**		**LM**		**PLS-GLM**		**NN**	
Predictors	# metrics	# entries	# entries	R^2	$\bar{E}$	R^2	$\bar{E}$	R^2	$\bar{E}$	R^2	$\bar{E}$
B	2	13,851	9,233	0.49	0.26	0.50	0.26	0,49	0,26	0.50	0.26
B + Radio	5	8,210	5,472	0.75	0.19	0.72	0.19	0,74	0,18	0.73	0.19
B + RAN	6	9,340	6,201	0.72	0.17	0.59	0.23	0,60	0,21	0.62	0.21
B + Context	5	10,710	7,140	0.59	0.24	0.57	0.25	0,60	0,24	0.58	0.25
B + Radio + RAN	9	5,872	3,903	0.87	0.11	0.79	0.16	0,81	0,15	0.79	0.16
B + Radio + Context	8	7,870	5,246	0.82	0.15	0.74	0.19	0,77	0,17	0.75	0.19
B + RAN + Context	9	7,320	4,863	0.86	0.13	0.67	0.21	0,72	0,19	0.73	0.19
B + Radio + RAN + Context	12	5,682	3,777	0.89	0.10	0.79	0.17	0,82	0,14	0.81	0.17

count more interactions between predictor variables. However, it also suffers from weaknesses inherited of the GLM method, especially the need to remove outliers from the predictors before fitting the model.

5.3.4. Artificial Neural Networks (NN)

We use a Feed-Forward Neural Network [41], which contains a single hidden layer with ten neurons. We uses at most 400 iterations and always performed a quick preliminary weight decay optimization. For each model, all feature vectors were fed to the neural network like for the other machine learning algorithms.

The principal advantage of this learning technique is its ability to fit with complex correlations between predictor variables. Moreover, the design based on a single layer of this NN approach results in algorithms that are easy to code and fast to execute. However, NNs are more sensitive to overfitting than the otheralgorithms considered in this study.

5.4. Results

We present in Table 4 the performance of the ML techniques for increasingly rich predictors. The baseline predictor, noted B, corresponds to the UE category and the cell band, because both metrics are always available in smartphone. Then, we increase the amount of available information. We show in Figure 8 the Cumulative Density Function (CDF) of the error ratio for all the predictions of the RF algorithm. Figure 9 provide an illustration by a heatmap of the predicted throughput against the actual measured throughput for the RF algorithm.

We first analyze the performance of the ML algorithms. Every approach

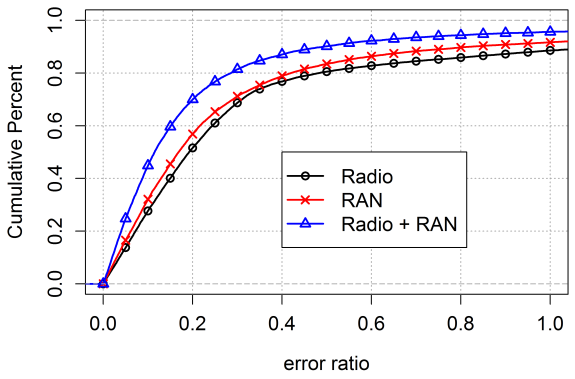

Figure 8: Cumulative distribution function of error rate

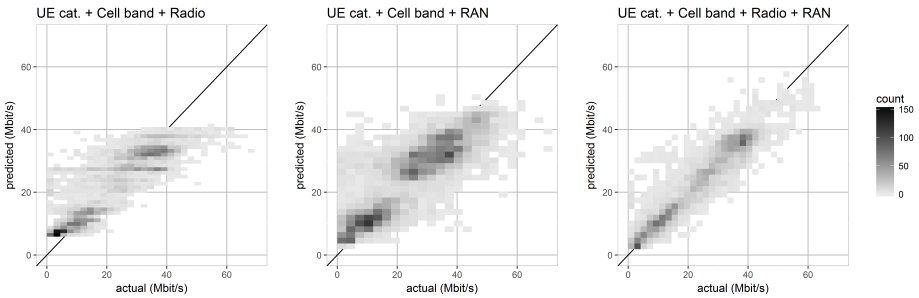

Figure 9: Actual versus predicted average throughput

succeeds in exploiting the increasing number of available metrics, which results in a growing accuracy. The RF peaks the performance with an R^2 value of 0.89 and a median error ratio at 0.10. The other approaches provides equivalent results. In particular, the PLS-GLM approaches improves the GLM algorithm by a low margin, which is insufficient to provide as accurate predictions as the RF algorithm. Our recommendation is therefore to use the RF algorithm due to its better performance. In comparison with other machine learning approaches, RF does not restrict to linear relationships, which enables the algorithm to efficiently capture the structural link between the predictor variables and the target variables. Of course, other machine learning algorithms, such as deeper neural network, may perform even better than RF in the future; however the

performance gains should not be significant since the choice of the used predictor variables is the most critical step of the prediction process in such a problem with structured data in input.

We then analyze the accuracy of the prediction. The results (especially a test validated coefficient of determination at 0.89 and a median error ratio at 0.10 for the most efficient prediction algorithms) are in the same performance range as much more sophisticated non-instantaneous techniques [19]. Our study thus reveals that instantaneous prediction based on data that can be available at the device and at the operator enables an accurate prediction. The *heatmap* presented in Figure 9 provides a graphical view of this accuracy for the RF algorithm. In Figure 8, two thirds of the predictions have an error ratio lower than 0.2, which means that the predicted throughput is between 0.8 and 1.2 times the actual throughput. This result validates our approach, which is to get an estimate of the average throughput of a download that is accurate enough to allow content providers to select a *class of service* for each end-user.

We finally analyze the quality of the various predictors. We have to find the balance between the prediction accuracy (the higher number of available metrics the better) and the system complexity (the lower number of metrics the easier to implement). First we show that the cell frequency band and the UE category do not enable an accurate prediction with our supervised learning technique. The context information improves performance, but the two main families of collected data that lead to a more accurate prediction are RAN and radio link (the coefficient of determination is 0.75 and 0.72 respectively). These results validate the correlation that we computed in Section 4. Second, RAN and radio input data are complementary since the combination of both increases the R^2 to 0.87 and limits the median error ratio to 0.11. These results are close to the best prediction we obtain with the best algorithm, when the predictor also includes the context information. In other words, the combination of the radio link information and the RAN information enables an accurate prediction. As we will discuss next, this result opens wide perspectives for the implementation of prediction-based adaptive services by content providers in mobile networks.

6. Discussion: Opportunities and Limitations

We discuss now the opportunities and limitations of our proposal. The achievable throughput of a connection over a cellular network depends on the performance of all the components that are involved in the transmission: the mobile device, the radio link, the cell capacity, the core network, and even the server of the content provider. At any given time, one of these components is the bottleneck that limits the data rate. A commonly accepted claim is that the network operator and the content provider provision the core network and Content Delivery Network (CDN) respectively so that the bottleneck is located in the so-called last-mile. If this is indeed the case, predicting the data rate in the last-mile is equivalent to predicting the end-to-end data rate.

At the same time, some researchers have also studied the case where the core network [45] or the CDN [46] are under-provisioned, in which case the bottleneck could not be in the last-mile. In this case, predicting the data rate on the wireless link is not enough to predict the end-to-end rate but such a prediction can still be exploited by combining it with information related to the status of the core network and the CDN. Typically, since content providers now use several CDNs [47] to deliver content, specific CDN monitoring solutions are emerging. It would therefore be possible to integrate the results of these monitoring solutions as inputs of our throughput prediction algorithm.

Another source of improvement for our algorithm is to use other types of information related to the mobile phone of the user. At the physical layer, the receiver's sensitivity and the transmission power are two variables that can impact the QoS. At the application layer, the buffer and even the performance of the OS are also critical elements. In our model, we take into account the category of the device, but we could go further into this analysis and evaluate if the prediction performance gap can be explained by other parameters related to the mobile phones. Typically, throughput optimization mechanisms such as Carrier Aggregation (CA) may be detected and integrated in our prediction process. The dataset that we used here does not collect such information; new

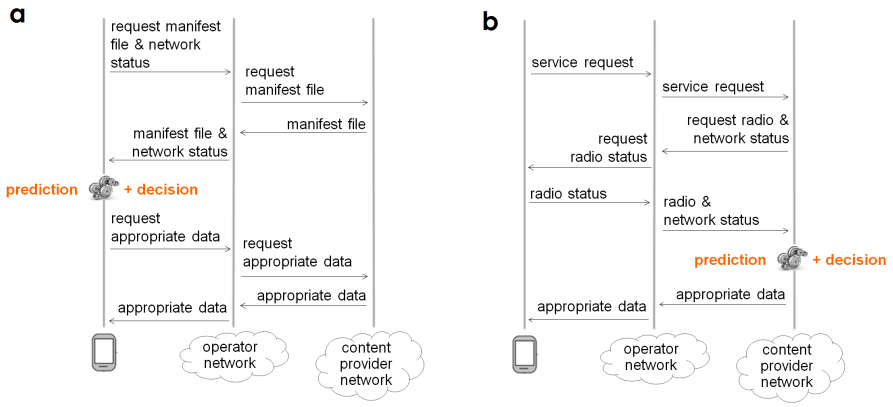

Figure 10: Implementation strategies. a: client-based pull delivery, b: server-based push delivery

studies should be conducted to get datasets that capture novel network and device technologies in the ever-evolving context of mobile networking.

Finally, a limitation of our study is the relatively poor availability of radio link information. Indeed, the measurements about radio link are accessed by the mobile device OS through specific APIs. Unfortunately, OS developers are increasingly restricting these OS APIs. Moreover, the documentation about these radio data from APIs is poor and their accuracy is sometimes controversial. Such uncertainty may come from the relatively low usage of these APIs by current mobile apps. In our proposal, we assume a full availability of the radio link information but this could require some more implementation efforts at the OS level. The Minimization of Drive Test (MDT) standard [48] is an opportunity to consider for solving this problem as It allows the network operator to access the radio link information for each subscriber.

7. Toward an Implementation

We address now a more general setting by considering our solution as one of the elements of a delivery chain involving content providers, network operators and mobile users. We claim that such a solution is a net positive for all the parties involved, justifying the corresponding costs.

We have shown that instantaneous, yet accurate enough, throughput prediction is possible but we have not yet explained how such a prediction could be shared and exploited. We provide here some details about the process we envision, shown in Figure 10. Given that a complete implementation is outside of the scope of this paper. We also discuss some possible future work in various areas.

We distinguish three main actors:

- The mobile user, who leverages the OS API to get radio link information. The mobile user can then grant access to this data to other actors. We have shown that context information is not a necessary input, so the risk of facing privacy issues due to sensible data is lower. Still, we do not hide the fact that sharing of information related to the immediate status of the mobile phone could lead to privacy issues.

- The content provider, who provides its service at different qualities. In systems like Dynamic Adaptive Streaming over HTTP (DASH), the provider describes these qualities in a *manifest file*, so that other actors are aware of the different options for the delivery.

- The network operator, who has to share information about the cell network status. Such information sharing does not exist today and one of the main contributions of our work is to highlight the benefits of sharing this information. In recent years, several proposals have already been discussed with similar goals. In particular, the MPEG DASH group has open an ad-hoc group to deal with Server and Network Assisted DASH (SAND) [49]. This proposal defines a DASH-Aware Network Element (DANE) [50], which is typically a set of APIs to let network operators offer updated information about the network status. Extending such proposal to cellular networks and to any mobile application should be considered.

Jiang et al. [51] have proposed a more general framework for sharing information among content providers, network operators and users. These three

actors should collaborate to improve the overall Quality of Experience (QoE) for the end-users. There are two main ways to implement adaptive delivery: client-based pull delivery, and server-based push delivery.

In client-based pull delivery (see Figure 10-a), the client should first get the manifest file from the content provider and the cell network status from the network operator. Then, the client combines its own information about the radio link and the cell network status to select the best option in the manifest file. The advantages include that the mobile does not reveal private information and that the processing is on the client side, alleviating the burden on the server. However, the information about the cellular network should be made available to any client and it could generate a non negligible amount of extra traffic since every new connection would mean a new request.

In server-based push delivery (see Figure 10-b), the content provider has access to the radio link from the mobile phone or from the network operator if MDT is implemented, and it retrieves cell network status information from the network operator. Then, based on the different service qualities, the content provider selects the best quality for the client. The advantages include that a given content provider can use one access to network operator to get information that can be used for many concurrent connections. However, it means that content providers can have access to radio link information of mobile phones. MDT supports the server-based push delivery concept as the content provider would have only one interlocutor that is the network operator that would provide both radio link and cell network status information.

8. Conclusion

In cellular networks, the last mile is generally the main bottleneck. To ensure an acceptable QoS level in such networks, service and content providers adopt adaptive delivery strategies, like in adaptive video streaming. Relying on a reliable throughput prediction is a key success factor in this context. Throughput prediction algorithms have been studied in the past but most of them are based

on a series of prior measurements. We argue that the prediction must not be dependent of any time consuming probing or data history storage. Applications or services require an instantaneous prediction of the throughput to ensure a satisfactory QoE, such as at the beginning of a streaming session. In addition, the prediction must be accurate for a sufficiently long time, which corresponds to the typical downloading time of a webpage or a video segment. Previous instantaneous history-less prediction algorithms address only short term predictions.

In this paper we present a throughput prediction solution that fulfills these aforementioned requirements. We investigate the inherent relation between the throughput and a collection of statistics that are already available before the beginning of a connection, at the client side and the network side. Leveraging different machine learning techniques, we show that an accurate prediction is possible by combining cellular link quality of the user and performance data of the access network. We thus raise the potential of assistance for content providers to handle adaptive techniques from the very beginning of their delivery. Nevertheless, sharing information between the parties represents the core condition of success. We finally propose some feasible implementation strategies to achieve this goal. The results presented in this paper represent a first step in the design of such operator-assisted strategies. We discuss some limitations of both our dataset and our prediction algorithms in Section 6. Further studies are needed to refine this proposal and analyze its effect on application performance. To complete this study, we are also interested in studying how the content providers could leverage an accurate historyless throughput prediction in their services, and how these service improvement could translate into a better QoE for the end-users.

Acknowledgment

We would like to thank all the people who accepted to take part to the test campaigns, the Orange Labs colleagues that provided the access to the RAN

data and the V3D platform, especially Fabrice Le Denmat. We also thank the mobile application skill center for their support. We thank Fabrice Clerot for his useful review and thank as well all the anonymous reviewers. This work is in part supported by the EU project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

References

[1] C. Hava Muntean, J. McManis, Fine grained content-based adaptation mechanism for providing high end-user quality of experience with adaptive hypermedia systems, in: ACM Proc. of the 15th Int. Conf. on World Wide Web WWW, 53–62, 2006.

[2] A. Vetro, C. Timmerer, Digital item adaptation: overview of standardization and research activities, IEEE Trans. Multimedia 7 (3) (2005) 418–426.

[3] R. K. P. Mok, W. Li, R. K. C. Chang, IRate: Initial Video Bitrate Selection System for HTTP Streaming, IEEE Journal on Selected Areas in Communications 34 (6) (2016) 1914–1928.

[4] M. Nagy, V. Singh, J. Ott, L. Eggert, Congestion control using FEC for conversational multimedia communication, in: Proc. of ACM Multimedia Systems Conference (MMSys), 191–202, 2014.

[5] W. Cai, R. Shea, C. Huang, K. Chen, J. Liu, V. C. M. Leung, C. Hsu, A Survey on Cloud Gaming: Future of Computer Games, IEEE Access 4 (2016) 7605–7620.

[6] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, M. Watson, A buffer-based approach to rate adaptation: Evidence from a large video streaming service, ACM SIGCOMM Computer Communication Review 44 (4) (2015) 187–198.

[7] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, M. Chiang, A scheduling framework for adaptive video delivery over cellular networks, in: ACM MobiCom, 2013.

[8] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, R. K. Sinha, Can accurate predictions improve video streaming in cellular networks?, in: ACM HotMobile, 2015.

[9] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, A. Terzis, CQIC: Revisiting Cross-Layer Congestion Control for Cellular Networks, in: ACM HotMobile Workshop, 2015.

[10] A. Samba, Y. Busnel, A. Blanc, P. Dooze, G. Simon, Instantaneous Throughput Prediction in Cellular Networks: Which Information Is Needed?, in: Integrated Network Management (IM), 2017 IFIP/IEEE International Symposium on, IEEE, 2017.

[11] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, SIGCOMM Comput. Commun. Rev. 27 (3).

[12] J. Padhye, V. Firoiu, D. F. Towsley, J. F. Kurose, Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation, IEEE/ACM Trans. Netw. 8 (2) (2000) 133–145, ISSN 1063-6692, doi:10.1109/90.842137, URL http://dx.doi.org/10.1109/90.842137.

[13] N. Cardwell, S. Savage, T. Anderson, Modeling TCP latency, in: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, IEEE, 1742–1751, 2000.

[14] B. Sikdar, S. Kalyanaraman, K. S. Vastola, Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK, IEEE/ACM Transactions On Networking 11 (6) (2003) 959–971.

[15] R. Prasad, C. Dovrolis, M. Murray, K. Claffy, Bandwidth estimation: metrics, measurement techniques, and tools, IEEE Network 17 (6) (2003) 27–35, ISSN 0890-8044, doi:10.1109/MNET.2003.1248658.

[16] N. Bui, F. Michelinakis, J. Widmer, A Model for Throughput Prediction for Mobile Users, in: European Wireless Conference, 2014.

[17] F. Ren, C. Lin, Modeling and Improving TCP Performance over Cellular Link with Variable Bandwidth, IEEE Transactions on Mobile Computing 10 (8) (2011) 1057–1070, ISSN 1536-1233, doi:10.1109/TMC.2010.234.

[18] Q. He, C. Dovrolis, M. Ammar, On the predictability of large transfer TCP throughput, in: ACM SIGCOMM Computer Communication Review, vol. 35, ACM, 145–156, 2005.

[19] M. Mirza, J. Sommers, P. Barford, X. Zhu, A Machine Learning Approach to TCP Throughput Prediction, in: ACM Sigmetrics conference, 2007.

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

[21] J. Jiang, V. Sekar, H. Zhang, Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive, in: Proceedings of the 8th international conference on Emerging networking experiments and technologies, ACM, 97–108, 2012.

[22] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, D. Oran, Probe and adapt: Rate adaptation for http video streaming at scale, IEEE Journal on Selected Areas in Communications 32 (4) (2014) 719–733.

[23] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, B. Sinopoli, Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction, in: Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference, ACM, 272–285, 2016.

[24] J. Yao, S. S. Kanhere, M. Hassan, An empirical study of bandwidth predictability in mobile computing, in: ACM WINTECH workshop, 2008.

[25] X. Xie, X. Zhang, S. Kumar, L. E. Li, piStream: Physical layer informed adaptive video streaming over LTE, in: ACM MobiCom, 2015.

[26] S. Lederer, C. Müller, C. Timmerer, C. Concolato, J. L. Feuvre, K. Fliegel, Distributed DASH dataset, in: Proc. of ACM Multimedia Systems Conference (MMSys), 131–135, 2013.

[27] EQual One - http://www.v3d.fr/solution/equal-one/, Vision 360 Degres (V3D), 2016.

[28] T. Everts, The average web page is 3MB. How much should we care? - https://speedcurve.com/blog/web-performance-page-bloat/, Speed Matters, 2017.

[29] Recommended upload encoding settings - https://support.google.com/youtube/answer/1722171, Youtube, 2018.

[30] 3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities (Release 14), Technical Specification 36.

[31] J. Postel, Transmission control protocol .

[32] X. Hu, X. Li, E. C. H. Ngai, V. C. M. Leung, P. Kruchten, Multidimensional context-aware social network architecture for mobile crowdsensing, IEEE Communications Mag. 52 (6) (2014) 78–87.

[33] S. Hahn, D. Gotz, S. Lohmuller, L. Schmelz, A. Eisenblätter, T. Kürner, Classification of Cells Based on Mobile Network Context Information for the Management of SON Systems, in: IEEE VTC Spring, 2015.

[34] S. Buuren, K. Groothuis-Oudshoorn, mice: Multivariate imputation by chained equations in R, Journal of statistical software 45 (3).

[35] A. Samba, Linkspotter, https://github.com/sambaala/linkspotter, 2017.

[36] R Core Team, R: A Language and Environment for Statistical Computing,

R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/, 2016.

[37] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, P. C. Sabeti, Detecting novel associations in large data sets, science 334 (6062) (2011) 1518–1524.

[38] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[39] P. McCullagh, Generalized linear models, European Journal of Operational Research 16 (3) (1984) 285–292.

[40] P. Geladi, B. R. Kowalski, Partial least-squares regression: a tutorial, Analytica chimica acta 185 (1986) 1–17.

[41] W. N. Venables, B. D. Ripley, Modern applied statistics with S-PLUS, Springer Science & Business Media, 2013.

[42] A. Liaw, M. Wiener, Classification and Regression by randomForest, R News 2 (3) (2002) 18–22, URL http://CRAN.R-project.org/doc/Rnews/.

[43] B.-H. Mevik, R. Wehrens, K. H. Liland, pls: Partial Least Squares and Principal Component regression, URL https://CRAN.R-project.org/package=pls, r package version 2.4-3, 2013.

[44] B. Efron, R. J. Tibshirani, An introduction to the bootstrap, CRC press, 1994.

[45] S. B. H. Said, M. R. Sama, K. Guillouard, L. Suciu, G. Simon, X. Lagrange, J.-M. Bonnin, New Control Plane in 3GPP LTE/EPC Architecture for On-Demand Connectivity Service, in: IEEE CloudNet, 2013.

[46] J. Liu, G. Simon, C. Rosenberg, G. Texier, Optimal Delivery of Rate-Adaptive Streams in Underprovisioned Networks, IEEE Journal on Selected Areas in Communications .

[47] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, Z.-L.

Zhang, Unreeling netflix: Understanding and improving multi-CDN movie delivery, in: IEEE INFOCOM, 2012.

[48] ETSI, Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Subscriber and equipment trace; Trace data definition and management, 3GPP TS 32.423 version 13.0.0 Release 13 .

[49] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, J. Famaey, Enhancing MPEG DASH performance via server and network assistance .

[50] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, S. Mascolo, Design and Experimental Evaluation of Network-assisted Strategies for HTTP Adaptive Streaming, in: MMSys, 2016.

[51] J. Jiang, X. Liu, V. Sekar, I. Stoica, H. Zhang, EONA: Experience-Oriented Network Architecture, in: ACM HotNet, 2014.