Lightweight Mobile Bandwidth Availability Measurement

Foivos Michelinakis[12], Nicola Bui[12], Guido Fioravantti[12],
Joerg Widmer[1], Fabian Kaup[3], David Hausheer[3]
[1]IMDEA Networks Institute, [2]Universidad Carlos III de Madrid,
[3]P2P Systems Engineering, Technische Universität Darmstadt
{foivos.michelinakis, nicola.bui, guido.fioravantti, joerg.widmer}@imdea.org,
{fkaup, hausheer}@ps.tu-darmstadt.de

***Abstract*—Mobile data traffic is increasing rapidly and wireless spectrum is becoming a more and more scarce resource. This makes it highly important to operate the mobile network efficiently. In this paper we are proposing a novel lightweight measurement technique that can be used as a basis for advanced resource optimization algorithms to be run on mobile phones. Our main idea leverages an original packet dispersion based, technique to estimate both per user capacity and asymptotic dispersion rate. This allows passive measurements using only existing mobile traffic. Our technique is able to efficiently filter outliers introduced by mobile network schedulers. In order to verify the feasibility of our measurement technique, we run a week-long measurement campaign spanning two cities in two countries, different radio technologies, and covering all times of the day. The campaign demonstrates that our technique is effective even if it is provided only with a small fraction of the exchanged packets of a flow. The only requirement for the input data is that it should consist of a few consecutive packets that are gathered periodically. This makes the measurement algorithm a good candidate for inclusion in OS libraries to allow for advanced resource optimization and application-level traffic scheduling, based on current and predicted future user capacity.**

***Keywords*—Mobile bandwidth measurement, Packet dispersion**

I. Introduction

Even though spectrum efficiency is improving thanks to the fifth generation [17] of mobile networks, the wireless medium is becoming a scarcer and scarcer resource, due to the ever increasing demand for mobile communication. Recently, a number of papers addressed improved resource allocation mechanisms based on bandwidth availability prediction techniques. For instance, [11], [1], [3] propose to use resources when they are more abundant and cheap, and to refrain from or to limit communication when it is more expensive (e.g., lower spectral efficiency, higher congestion, etc.) by exploiting perfect knowledge of the future bandwidth.

Recently [2], we surveyed the state of the art on mobile bandwidth prediction techniques and built a model for both short and medium to long term prediction errors in order to be able to quantify the impact of prediction uncertainties in resource allocation. By developing this model, we noticed that most short term prediction techniques [13], [15] rely on time series filtering solutions, such as moving average and autoregressive (ARMA) or autoregressive conditional heteroskedasticity (ARCH) modeling. Thus, in order to allocate resources on a given time granularity, prediction must be available with the same granularity and, consequently, mobiles must be able to measure bandwidth availability with the same granularity.

Mobile bandwidth measurement is a well investigated topic in the literature, but, to the best of our knowledge, no lightweight or passive technique allows mobiles to collect frequent measures of their available bandwidth. To fill this gap, this paper proposes a simple technique which is able to measure the fast variations of the per user capacity and, from those, the expected end-to-end throughput. In order to do so we adapt packet train dispersion techniques by applying an adaptive filtering mechanism, which we show is effective in removing the impact of outliers due to bursty arrival and jitter, which are very prevalent in mobile environments. We validate the effectiveness of the solution through an extensive measurement campaign: our technique can achieve an accurate throughput estimate with as few as 5 % of the packets needed by other solutions, while making an error smaller than 20 %.

Our goal is to provide a simple tool that evaluates passively or with minimum impact the per user capacity variations over time in a mobile environment. This enables filter based prediction techniques and, consequently, prediction based resource allocation optimization. Source code for the tool can be found in the repository of the EU project eCOUSIN[1].

In the following sections we propose a lightweight measurement technique of the per user cell capacity. Our proposal adapts earlier packet train dispersion techniques and allows to collect reliable measurements on a mobile device despite the complexities introduced by the wireless link. Also, we have evaluated our technique on actual mobile network data collected during a measurement campaign.

The rest of the paper is structured as follows: related work and some mobile network fundamentals are discussed in Sections II and III respectively, we present our measurement technique in Section IV, while in Section V we describe how we collected the data needed to validate this technique. The obtained results and their discussion are in Section VI. Finally, Section VII summarizes our conclusions.

[1] https://ecousin.cms.orange-labs.fr/sites/ecousin/files/lightmeasure.zip

ISBN 978-3-901882-68-5 © 2015 IFIP

II. Related Work

A number of approaches exist to estimate mobile bandwidth. The most popular of which is Ookla's mobile application, Speedtest [12], which computes the maximum end-to-end throughput achievable by a single long lived TCP connection (according to our tests the measurement lasts for either 20 seconds or after 30 MB have been downloaded, whichever happens first) with the closest measurement server. Then, it derives throughput samples and aggregates them into 20 bins (each one has about 5% of the samples), applies some post processing to remove measurement artifacts and, finally, estimates the average of the bins. Huang et al. [6] proposed to use 3 parallel TCP connections in order to remove the effects of packet losses, TCP receive window limitations and overloaded servers, while ignoring any data collected during the slow-start phase of TCP. The calculated throughput is given by the median of the collected samples, in order to reduce the effect of outliers. Recently, Xu et al. [19] analyzed the use of UDP to compute the end-to-end throughput availability, also accounting for packet interarrival times and the impact of mobile scheduling. All these techniques are active, use long data transfers and thus, incur a high overhead.

Conversely, passive monitoring techniques aim at estimating similar information by analyzing ongoing mobile communications, without triggering any dedicated activity. Gerber et al. [5] achieved quite accurate results just by relying on selected types of applications (i.e., video streaming), which provide more reliable throughput measurements as they are more likely to exploit the full cell capacity. In order to study transport protocols in LTE, [7] developed a passive measurement scheme, which monitors the sending rate over a given time window that ensures the full exploitation of the available bandwidth. PROTEUS [18] combines passive monitoring with linear prediction to estimate the achievable throughput. Other worth mentioning solutions in this category are [14], where the authors try to identify bottleneck links in the core network of an operator by conducting large scale passive measurements of TCP performance parameters and citesvoboda2008analysis, where network "footprints" (generated by counting the number of packets and the number of retransmissions of all the users of a network) were used to identify capacity bottleneck links. However, these solutions cannot be directly applied to mobiles.

We conclude that none of the aforementioned solutions allow for frequent throughput measurements, nor do they provide estimates of the per user cell bandwidth availability on the client side (mobile device) to allow for effective bandwidth prediction and resource allocation.

Lai [10] attempts to actively measure link bandwidths of a path by taking advantage of the packet pair property of FIFO-queuing networks. Dovrolis [4] further refines the packet pair technique and demonstrates that packet pair dispersion rate has a multimodal distribution, whose modes in turn depend on the capacity and the cross traffic at each of the links composing the sender-receiver path. Also, the authors devise a method to estimate the capacity of the bottleneck link in the path, based on the fact that the average throughput measured by packet trains converges to the asymptotic dispersion rate, from which an estimate of the bottleneck capacity can be computed. As we will discuss later though, it is unsuitable for use over mobile networks. CapProbe [8] proposed a technique based on packet pairs dispersion and delays to devise a reliable bandwidth estimation technique, aimed at mobile networks. Both techniques are meant to measure the capacity of the bottleneck link of a path. Instead, we are interested in measuring the per user available bandwidth at a given moment.

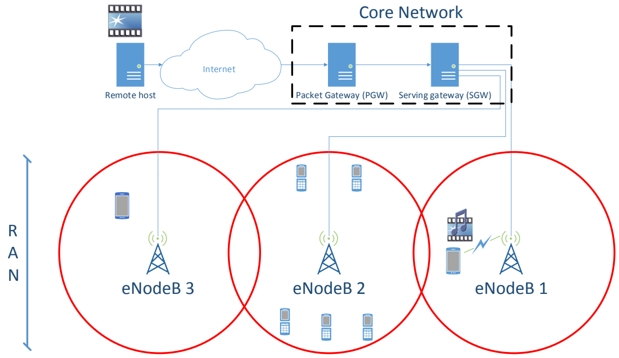


Fig. 1: Some of the LTE network components that a file has to traverse in order to reach a mobile client.

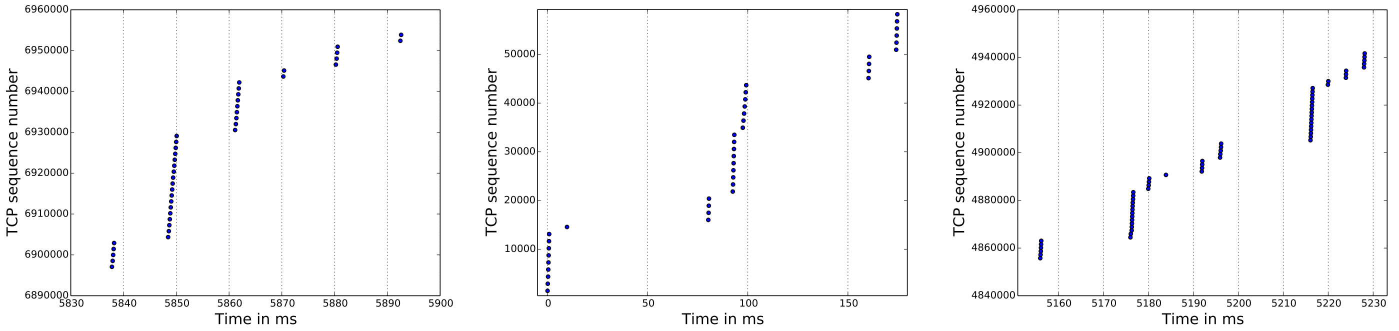
In this paper we use the relationship between the per user capacity C_U, and the end-to-end throughput R, to get an estimate of the latter using sparse samples of the former, which is measured using our proposed packet dispersion technique.

III. Mobile Networks characteristics

In this section we provide a brief overview of the components and characteristics of mobile networks that have an effect on bandwidth measurement. In the rest of the paper, we will use terminology and network architecture components of LTE, but the ideas and the algorithm can be applied to any recent mobile network technology like 3G.

The user equipment (UE), which can be any device with mobile communication capabilities, connects to the operator network through any of the multiple base stations (BS) that the operator controls, as shown in Fig. 1. BSs are in turn connected to the core network (CN) of the operator. This set of antennas can be collectively called Radio Access Network (RAN) and are the interface between the UE and the operator.

The transmission of data from the BS to the multiple UEs connected to it, is regulated by a scheduler, which allocates resources and periodically transmits packets to the associated UEs. This period, called Transmission Time Interval, (TTI) largely differs among mobile telecommunications systems, with more recent technologies having lower values. It can be as short as 1 ms for LTE or at least 10 ms for UMTS. Thus, the UEs receive data in a way such that a burst of data is transmitted to them, during TTIs that they have been allocated resources and receive nothing during TTIs that they have not been allocated resources. The scheduling process is usually based on a fairness scheme that takes into account the data needs and channel quality of all the UEs served by the same BS. A very popular such scheme is the "proportionally fair" [9]. It tries to weight the past allocation of resources and the current potential throughput of all the competing users. That way it finds a balance between providing adequate



(a) Link saturation traffic over LTE during the steady state of a TCP flow.

(b) Arrival of the first packets of a TCP flow over LTE.

(c) Some packets may be registered with a noticeable delay.

Fig. 2: Time-sequence graphs presenting the arrival of packets to a smartphone, as they were captured by the traffic sniffing tool tcpdump. The time values represent time since the first packet of the download arrived and represent when the related packets were captured by tcpdump, not their actual arrival time at the device.

resources to all users, regardless of their channel quality, and maximizing the overall throughput of the base station.

When a packet is transmitted to a UE, it travels from the Internet to the operator's core network which forwards it to the base station that the UE is connected to. The packet is then stored at the base station in a buffer dedicated to the recipient UE. The packet remains to the dedicated buffer until the scheduler decides to allocate resources to the recipient UE. Upon allocation and depending on the signal quality, it is either grouped alongside other packets present in the buffer to a Transport Block (TB) or, in cases of very bad signal and/or small amount of allocated resources, a segment of it is encapsulated in a TB. The TB is then sent to the UE.

The above are illustrated in Fig. 2a, which shows the arrival of packets to an LTE smartphone, as it was captured by the sniffing tool tcpdump. In this experiment we are saturating the link and observe its behavior during TCP steady state. Note that the TTI of LTE is fixed to 1 millisecond. It is easily observable that the packets arrive in groups that have about the same duration as the TTI. Between these groups of packets, the smartphone is not allocated resources, thus nothing is received. The size and time spacing of the groups depend on the the channel quality of the UE and the congestion of the BS.

A. *Packet pairs issue*

The previous characteristics of the mobile networks make the use of traditional packet pair techniques unfeasible. Any two packets that would make a packet pair are in either of the following cases.

Transmitted in the same TB. In this case the packets arrive more or less at the same time to the UE, since all the information included in the TB is transmitted in parallel using multiple carrier frequencies. The lower protocol layers of the UE ensure that they are delivered to the higher layers in the right order. Consequently, sniffing tools like tcpdump perceive them as arriving with a tiny time difference, in the order of a few hundreds of microseconds. A bandwidth estimation based on these packet pairs would greatly over-estimate the real value of the bandwidth.

Transmitted in different TBs. In this case, the packet pair consists of the last packet of a TB and the first packet of the following TB. Thus, the bandwidth value is greatly underestimated, since the measured dispersion is the dispersion between the TBs and based on our experiments allocation of consecutive TBs to the same UE is infrequent. If there is exactly one packet per TB, then an accurate estimation is possible, but we observed that in the majority of the cases each TB contains multiple packets.

B. *Measurement artifacts*

In our traces we frequently observed measurement artifacts that are unrelated to the scheduler and can are due to following reasons.

Small congestion window values during the slow start: the servers that transmit data over TCP send bursts of packets to the client and wait for the related acknowledgments before sending more. This behavior is very prominent during the slow start phase of the transmission when the congestion window has small values. The gap in the transmission at the server side may cause an analogous gap in the transmission at the base station. During this time, the base station is not sending data to the recipient UE, because there are not data in the dedicated buffer. This is visible in Fig. 2b, which illustrates the delivery of the first packets of a TCP flow over LTE. In two occasions, consecutive TBs are received with a delay in the order of tens of ms. We also observe in this example, that the total number of packets delivered in the groups that arrive at about 75 ms is bigger than the number of packets in the first set of groups (the second group has just one packet) at 0 ms. This is caused by the exponential growth of the congestion window. Eventually, the congestion window is big enough that the we observe a continuous stream of incoming packets and this effect diminishes. Since the Round Trip Time is bigger in 3G networks, the impact of this TCP behavior is slightly more pronounced.

Weak or busy phone hardware: we observed that the exact timing of packet arrival is affected by the capabilities of the phone and the capture software[2]. Usually packets are registered in the trace with a delay in the range of a few ms. More powerful phones are a bit less affected, but even in this case, the delay shows slight variations. Since this delay is very small, it is not significantly affecting our technique, whose adaptive and statistical nature tries to countermeasure it. The same effect is reported in [19], where it is attributed to the polling frequency of the OS to the NIC driver. Also, it is quite common for packets to be delivered to the phone but not delivered to the higher layers until several milliseconds later, alongside all the other packets that have been received in the meantime. This is usually observed in cases of high available bandwidth and/or high CPU utilization. This behavior is very evident in Fig. 2c, which depicts the TCP steady state of a 3G download. According to the server side trace of this download, the server transmitted all the packets that are visible in the figure almost "back-to-back". Also, the phone trace showed a steady rate in the delivery of packets. But at times 5175 and 5215 ms we observe a gap in the delivery of packets and then the delivery of an impossibly big group. Packets were actually delivered during these gaps were registered all together when the CPU was able to process them.

C. Packet trains issue

Packet trains are also problematic for the following reasons. They can not be used in a **passive scenarios** because the server transmits packets on the receipt of ACKs and the application requirements, so the trains will have variable length.
The number of packets in each **TB may be different**, which results in similar problems to the ones described in the "packet pair" scenario. On one occasion all the packets will be transfered in the same TB and on another in multiple TBs.

It is clear that long-established packet dispersion techniques that were developed to detect the bottleneck link capacity at wired networks are not suitable for mobile networks, especially in regards to detecting the per user capacity. In the sequel, we will present the necessary modifications to this approach to make it provide bandwidth estimations in mobile scenarios.

IV. Mobile Bandwidth Estimation

In a mobile network, C_U is the per user capacity of the link between the BS and the UE. This is the last hop of the downlink path and its capacity is dependent on the cell congestion and the channel quality. Consequently, it is usually the link of a path with the lowest available bandwidth, that also contributes the most to the delay. With R we denote the end-to-end TCP throughput achieved by mobile applications, which depends on the capacities and the cross traffic of all the links in the downlink path. The end-to-end TCP throughput is primarily determined by the link with the minimum available bandwidth, which in a mobile scenario is usually the RAN.

Fig. 3 illustrates the packet dispersion due to the transmission over links with different capacities. Initially, (1) the server sends a burst of IP packets (A-H in the example) back to back. The number of packets in the burst varies since it depends on a number of factors like the state of TCP connection and the specifics of the application and the server that generates it. Subsequently, (2) the base station (eNodeB) receives the packets, which have suffered variable delays due to the different link capacities and cross traffic encountered along the path. When the scheduler allocates a TB (marked with horizontal arrows in the plot) to the receiving UE (3), as many packets as possible are encapsulated in it. Therefore, all the packets that are scheduled together arrive within the same TTI at the UE. In fact, the inter-packet interval can be greatly reduced (packets A and B) or greatly magnified (packets B and C). Similar effects are observed in various mobile technologies.

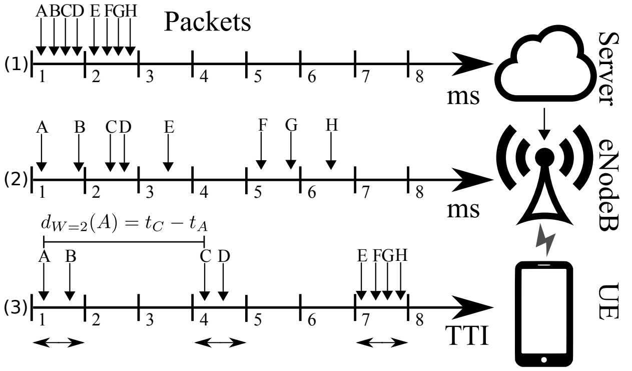


Fig. 3: Dispersion of IP packets over the Internet. First, they are sent back-to-back from the server (1). After experiencing disperion on the Internet, they arrive on the BS (eNodeB) (2). Finally, they are received in groups by the UE (3). The timelines (1-3) in the figure happen sequentially, one after the other, not in parallel. The horizontal arrows represent a TB allocated to the recipient UE.

Considering the set of "back-to-back" transmitted packets crossing the two-link path in Fig. 3, the first link (backhaul) and the second link (cellular) are characterized by bandwidth C_B and C_U respectively. If $C_B > C_U$, the set of packets arrives at the second link with a delay which is inversely proportional to C_B and shorter than the average time needed for the second link to serve all but the last packet. In other words, the arrival rate is higher than the departing rate at the base station, thus the dispersion of the set is caused by the last link. Also, depending on the scheduling strategy, the set may be served within the same transport block or multiple transport blocks by the BS. Conversely, if $C_B < C_U$ the set of packets arrives to the second link separated by a delay which is longer than the average serving time of the BS. We thus have three cases (excluding the problematic cases of section III): *i*) bursty arrival [7], [19] (e.g.: set of packets E-F), if $C_B > C_U$ and packets are in the same transport block, *ii*) last hop capacity if $C_B > C_U$ and packets are in different transport blocks (e.g.: set of packets A-D), or *iii*) lowest hop capacity if $C_B < C_U$.

In order to estimate C_U, we have to filter both *i*) and *iii*) cases, as well as take into account the behavior of sets of packets when transmitted over mobile networks as presented in section III. In brief, our approach has two components: a) generating bandwidth estimation samples which are not significantly affected by the above and b) the statistical processing of those samples in order obtain a C_U value. A high level overview of the technique is presented in Table I.

[2] http://www.tcpdump.org/faq.html#q8 [Last access: 2015-03-24]

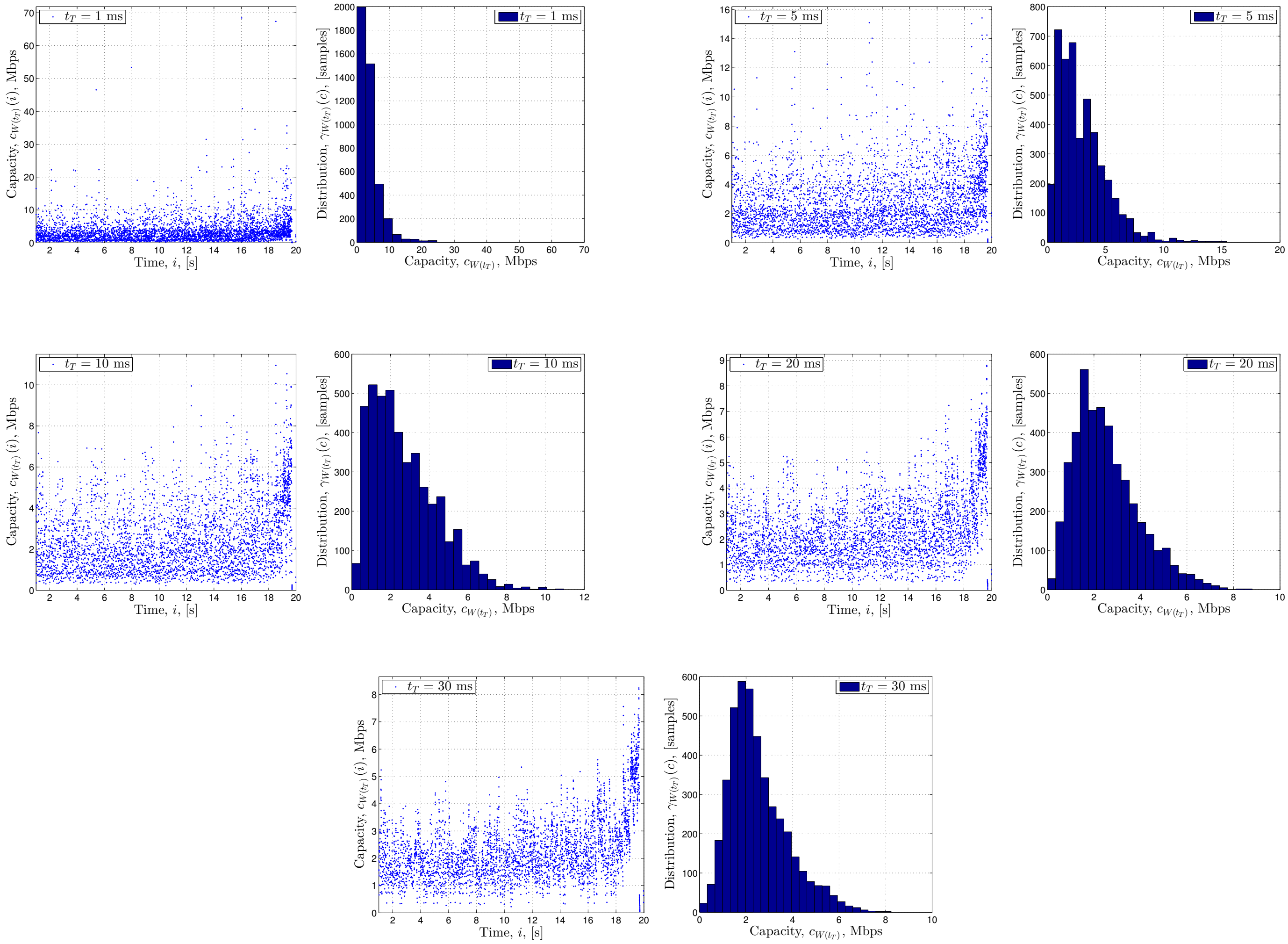


Fig. 4: Scatterplots of $c_{\hat{W}}$ (left of each pair) and histograms of $\gamma_{\hat{W}}$ (right of each pair) computed for $t_T = \{1, 5, 10, 20, 30\}$ ms from left to right. When the dispersion time is computed on windows larger than the TTI, $t_T > t_S$, the dispersion time distribution gets more stable.

Description of step	Related metrics	Notes
Choice of parameters.	t_T, bin size and f (if applicable) are chosen.	The optimal t_T for various Technologies can be found at Table II.
Generation of estimation samples that are not affected by type *i*) arrivals.	For each captured packet i, we define a dispersion time $d_W(i)$ and then a per user dispersion rate $c_W(i)$.	The duration of the $d_W(i)$ is different per packet. The minimum duration is given by Eq. (6), which ensures that at least two packets belonging to different TBs are taken into account. For some packets towards the end of a flow (or bin) it is impossible to generate those metrics, since the minimum $d_W(i)$ is bigger than the time between their arrival and the end of the flow (or bin).
Selection of samples that are not affected by type *iii*) arrivals.	Generation of the dispersion rate distribution. Its maximum reflects the per user capacity.	Values smaller than the maximum can be attributed to bottleneck links other than the BS-UE link. Asymptotic dispersion rate is given by the average of the distribution [4].

TABLE I: The different steps of the technique

A. *Bandwidth Estimation Samples*

The input data for our passive measurement tool are the timestamps and sizes of all the received packets of a smartphone. This information can be collected on the OS level by monitoring the stack. In our experiments, we used rooted Android smartphones and tcpdump to capture all the incoming traffic. Ultimately this functionality could be included in the mobile OS as an on-demand lightweight measurement service.

We consider a set of N packets sent back-to-back from a server and received at the UE so that the i-th packet is received at time t_i, with $i = \{1, \ldots, N\}$. A key metric used by our algorithm is the "inter-packet interval", the time difference between the arrival of two consecutive packets $(t_{i+1} - t_i)$. Obviously, in a group containing N packets, there are $N - 1$ intervals. W represents the number of such intervals that we take into account when we generate the bandwidth estimation samples. For each packet in the set we define the dispersion time $d_W(i) = t_{i+W} - t_i$, and the per user dispersion rate $c_W(i) = (\sum_{j=i}^{i+W-1} L_j)/d_W(i)$, for a given value of W, where L_i is the length of i-th packet.

In detail, the $c_W(i)$ value of packet i is derived by adding the sizes of W consecutive packets, starting from i and then dividing by the time duration of W consecutive inter-packet intervals, starting from $[t_{i+1} - t_i]$. Packet $i + W$ contributes only to the denominator. For example, in Fig. 3, $c_{W=2}(A)$ is computed by dividing the sum of sizes of the packets A and B by the dispersion time $d_{W=2}(A) = t_C - t_A$.

Subsequently, we define $\delta_W(d)$ and $\gamma_W(c)$ as the statistical distributions of the packet dispersion time and the dispersion rate, respectively. Note that the three arrival cases above contribute to those distributions in different ways: arrivals of type *i*) cause a tiny $d_W(i)$ and, thus, influencing the right part $\gamma_W(c)$ (over-estimation of C_U), while the left part of it is influenced mostly from type *iii*) events, which show larger $d_W(i)$ (under-estimation of C_U). To better visualize what is discussed next, Fig. 4 shows a set of scatterplots of $c_W(i)$ and their related histograms of $\gamma_W(c)$ computed on a single download performed using the Speedtest application [12] over a HSPA connection.

First, we may want to try to limit the impact of type *i*) arrivals by setting W appropriately: the idea is to include in each measurement packets belonging to different TBs in order to make sure that the highest throughput $c_W(i)$, we can measure is only related to the cell capacity and not to bursty packet arrivals, as it would have happened had we chosen $W = 1$ in the example of Fig. 3. In order to achieve that, it is sufficient to study groups that contain $\hat{W}(i)$ intervals so that the minimum dispersion time is longer than the maximum TTI, abbreviated t_S:

$$\hat{W}(i) = \{\min(W) | \min(d_W(i)) > t_S\}; \tag{1}$$

in fact, this guarantees that at least two packets within the $\hat{W}(i)$ window are scheduled in two different transport blocks since $t_{i+\hat{W}(i)} - t_i = d_{\hat{W}(i)}(i) > t_S$. In other words, we are averaging the burstiness over two transport blocks. An effect of Eq. (1), is that each packet i has a different W value, depending on the spacing of packets that were received after it.

It is important to select the minimum value of W for the creation of the $c_W(i)$ value for packet i that has the property $\min(d_W(i)) > t_S$. As discussed in section III, the "slow start" behavior of TCP introduces noticeable gaps in packet delivery. Thus, samples that include these gaps in their calculation of $d_W(i)$, generate $c_W(i)$ values that are significantly smaller and not representative of the C_U. A high value of W increases the probability of a sample to include such gaps.

B. Statistical Processing Of The Samples

Now that type *i*) events are filtered, the minimum dispersion time $\min d_{\hat{W}(i)}$ cannot be smaller than the minimum time needed for the set of packets to cross the last link, which corresponds to the maximum per user cell capacity. Thus, C_U can be found as the maximum of the distribution $\gamma_{\hat{W}}(c)$:

$$C_U = \max_c \gamma_{\hat{W}}(c), \tag{2}$$

note that, with Eq. (1) we are filtering the effect of type *i*) arrivals (min) and with Eq. (2) the delays introduced by type *iii*) arrivals (max) and server behavior.

Ideally, we would like to sample $c_{\hat{W}}$ until the $\gamma_{\hat{W}}(c)$ is stable, but C_U is varying because of both user movements and fast fading, hence we can only obtain an estimate $\tilde{c}^{(K)}_{\hat{W}}$ of it from a set of K consecutive dispersion time samples. Although estimating the distribution from a limited number of samples reduces the accuracy of our measurement, we can at least guarantee that we are not overestimating C_U:

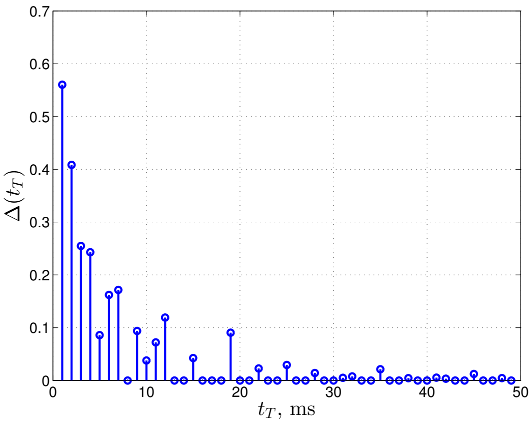


Fig. 5: Ratio $\Delta(t_T)$, varying $t_T \in [2, \ldots, 50]$ ms. The measurements get stable from $t_T > t_S = 10$ ms.

$$\tilde{C_U}^{(K)} = \max_c \tilde{\gamma}^{(K)}_{\hat{W}}(c) \leq \max_c \gamma_{\hat{W}}(c) = C_U, \tag{3}$$

where $\tilde{\gamma}^{(K)}_{\hat{W}}(c)$ is the distribution of $\tilde{c}^{(K)}_{\hat{W}}$ obtained using K dispersion samples. This follows from the probability of the distribution of a sampled random process to contain the maximum of the theoretical distribution of the process, which is increasing with the number of collected samples:

$$\lim_{K\to\infty} \tilde{C}^{(K)} = C_U. \tag{4}$$

Finally, with similar reasoning, the asymptotic dispersion rate, $R = \mathrm{E}[c_W]$ [4] and its sampled version $\tilde{R}^{(K)}$, tends to the actual value when the number samples tends to infinity:

$$\lim_{K\to\infty} \tilde{R}^{(K)} = R. \tag{5}$$

C. Bandwidth Measurement

This section describes the feasibility of lightweight active and passive measurements of per user capacity C_U and asymptotic dispersion rate R based on dispersion samples of packet sets. It also explores the effect different values of some parameters have on our technique. We compute the dispersion time by using an adaptive window $\hat{W}(i)$ intervals such that:

$$\hat{W}(t_T) = \arg\min_W(t_{i+W} - t_i > t_T), \tag{6}$$

where $t_T \in [1, \ldots, 50]$ ms, for all the values of t_T. So, each sample i is composed of all packets following i until the first packet which arrived after more than t_T ms after i. This allows to satisfy Eq. (1) a posteriori if the TTI duration is not known.

We exemplify the dispersion time in Fig. 4, based on data obtained by time-stamping the arrival time of the packets of a 6-MB HSPA download. This figure presents the evolution of the scatterplots of $c_{\hat{W}}$ and histograms of $\gamma_{\hat{W}}$ for various characteristic values of t_T.

During the slow start phase of a TCP connection an increasing number of packets are sent back to back from the

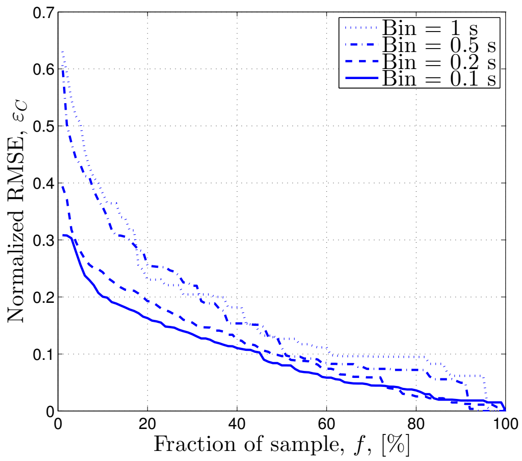


Fig. 6: Normalized root mean square error ε_C of the capacity estimate computed over a fraction $f = K/N$ of continuous samples for varying bin sizes ({0.1s, 0.2s, 0.5s, 1s}).

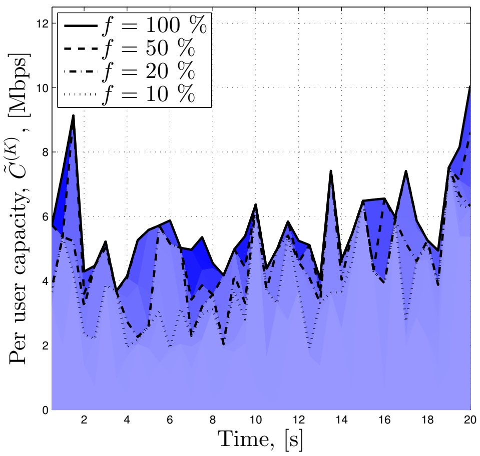


Fig. 7: Time plot of the capacity variation $C_U(t)$ computed every 500 ms and its different estimates computed with $f = \{10, 20, 50, 100\}$ %.

server, and after a few round trip times (RTT) the congestion window is large enough to allow the transmission of packet trains long enough to measure capacity as high as 100 Mbps. In fact, C_U should be proportional to the maximum number of packets that got scheduled in a single transport block and, if Eq. (1) is satisfied and $t_T > t_S$, the impact of outliers due to bursty arrivals is removed. With reference to Fig. 4, it can be seen that the maximum of the capacity distribution $\max_c \gamma_{\hat{W}(t_T)}(c)$ is approaching a stable value of about 10 Mbps when $t_T \geq 15$ ms. Due to limited space, we can not present the related plots of other downloads. Based on the rest of our dataset, a stable value is reached for values of t_T between 10 and 20 ms.

Moreover, Fig. 5 shows the stability of the maximum of the capacity distribution by plotting the ratio $\Delta(t_T)$, computed between the maximum of the distributions obtained with windows of t_T and t_{T-1}:

$$\Delta(t_T) = \frac{|C_{\hat{W}|t_T} - C_{\hat{W}|t_T-1}|}{C_{\hat{W}|t_T-1}}. \qquad (7)$$

Ideally, the ratio $\Delta(t_T)$ should stabilize to 0 as soon the scheduling outliers are filtered ($t_T > t_S$) and further increasing t_T should only make the distribution smoother. However, in actual experiments increasing t_T makes it more difficult to obtain a sample of the maximum capacity which is consistent over different transport blocks. In this preliminary example, we can see that $\Delta(t_T)$ becomes stable for $t_T > 20$ ms, which is in line with the HSPA TTI of $2 - 10$ ms.

Next, we divide the time duration of a download into fixed sized bins and we apply the above method taking into account only a percentage $f = K/N$ of consecutive capacity samples in each bin. Fig. 6 shows the normalized root mean square error, ε_C of the estimate by varying f:

$$\varepsilon_C = \sqrt{\frac{\sum_{\text{bins}}(\tilde{C}^{(K)} - \tilde{C})^2}{N_b \mathrm{E}[\tilde{C}]^2}}, \qquad (8)$$

where N_b is the number of bins in a flow. The computations have been repeated for different bin sizes varying in $\{1, 0.5, 0.2, 0.1\}$ seconds (dotted, dash-dotted, dashed and solid lines, respectively). It can be seen that the error decreases below 20 % when more than 20 % of the samples are used.

Also, Fig. 6 can be interpreted as the width of the distribution of the probability of an exact measurement using f % samples. In particular, it is easy to see that, if we use all the samples, the distribution should collapse into a delta function (zero width), while the fewer sample we use, the larger the distribution bell. Notably, this distribution should be positive for values larger than the actual measurement only, due to Eq. (3) that shows $\max_c \tilde{\gamma}^{(K)}_{\hat{W}}(c) \leq \max_c \gamma_{\hat{W}}(c)$. Thus, the real value can only be larger than the measured one. Since our technique estimates the maximum of the user capacity, it can be seen within the framework of extreme value theory [16]. We left the refinement of our technique according to this theory for future work.

To complete this preliminary evaluation of our measurement technique, Fig. 7 shows the variation of the per user capacity $C_U(t)$ measured every 500 ms and its estimates $\tilde{C}^{(K)}(t)$ computed with $f = K/N = \{10, 20, 50, 100\}$ % (dotted, dash-dotted, dashed and solid lines, respectively). Although with 10 % of samples the estimates are quite different from the actual capacity values, we will be showing next that it is possible to exploit these coarse estimates to obtain a sufficiently accurate asymptotic throughput estimate.

V. Measurement Campaign

In order to validate our measurement technique over many different scenarios and configurations, we organized a measurement campaign that covers two cities in two different countries, Madrid (Spain) and Darmstadt (Germany), for 24 hours a day lasting 7 days. During this time, 5 people per city moved around as they normally do, carrying one measuring device each and performing their usual tasks involving mobile networking on the measuring devices. In order to be able to compare results of both passive and active measurements, we also perform automated periodic file downloads.

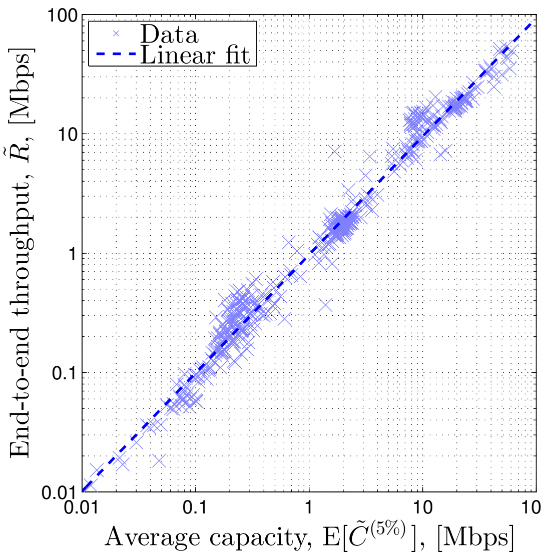


Fig. 8: Scatterplot of the average estimate of per user capacity $\mathrm{E}[\tilde{C}^{(K)}]$ computed over 5 % of the available information ($K = 5\ \% * N$) against the estimated end-to-end throughput measured using all available information. The dashed line shows the linear regression between the two quantities.

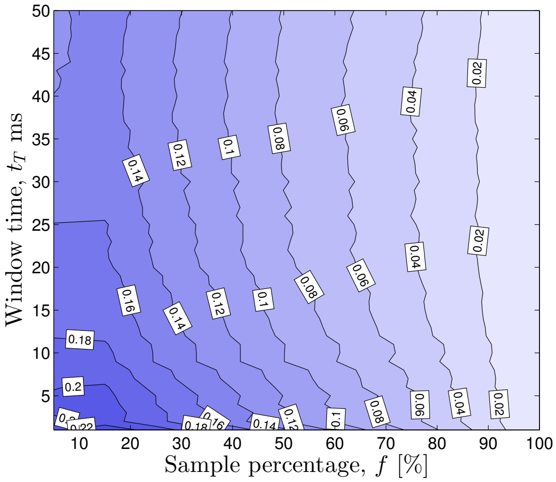


Fig. 9: Contour graph of ε_C varying t_T and f for a bin size of 200 ms.

All the devices were running a simple Android application, which was periodically sampling the available capacity by starting two download types: *short* downloads of 500 KB to study the TCP slow start phases and *long* downloads of 2 MB to measure TCP steady state throughput. The two types were organized in a sequence with a long download, preceded by two small downloads and later succeeded by another two. We use tcpdump on the measurement devices to monitor the arrival time and size of all incoming packets. The download sequence was repeated every 50 minutes. Additionally, we log other related phone parameters: GPS, cell ID, Channel Quality Indicators (ASU, dBm) and network technology (2G, 3G, LTE).

The phones used in the campaign were the following: 5 Nexus 5, located in Germany, and 4 Sony Xperia Miro and 1 Samsung Galaxy S3, located in Spain. Also, while the Nexus 5 phones are LTE capable, the other phones only support radio technologies up to HSPA.

VI. Results and Discussion

We verified our measurement technique by analyzing more than 3000 unique TCP flows extracted from the communication of the phones participating in the campaign. For each flow we compute the asymptotic dispersion rate $\tilde{R}$, using Eq. (5) accounting for the whole information available, and the per user capacity $\tilde{C}^{(K)}$ for different percentages of used information $f \in [1 - 100]$ % and different bin sizes (from 0.1 to 2 s).

The first and most important result, Fig. 8, shows a scatterplot where each point is obtained from a different flow for which the estimated average per user capacity is used as the abscissa and the end-to-end throughput as the ordinate. We also plot the linear regression between the two quantities as a dashed line in order to show that one is a good predictor of the other. We compute the per user capacity in each bin of the flow by using 5 % of the samples of the bin and we average over the bins, while the throughput is obtained using all the available information and therefore is considered to be the ground-truth for this comparison.

The figure is plotted in double logarithmic scale in order to emphasize that the relationship between R and C_U can be observed over all the measured connection rates. Although outliers are visible, we can obtain quite an accurate estimate of the end-to-end throughput by exploiting as few as 5 % of the packets sent during a TCP connection. This allows for quite an effective passive monitoring technique as, even by monitoring small data exchanges, it is possible to obtain frequent and accurate mobile bandwidth availability measure necessary for user throughput prediction and resource allocation.

Unfortunately, using low rate background traffic is impossible, because either the rate is incredibly low (about 4 packets over 100ms) or the APPs use the Google Cloud Messaging (GCM) service. In the case of GCM, if there is an update a couple of packets are sent just to generate a notification. When the user interacts with the notification, a larger number of packets are downloaded. In this scenario, we can use that download to get and estimation.

As a side note, our technique is also able to estimate fast per user capacity variations. However, it obtains a lower accuracy since a larger fraction of samples are needed to estimate the maximum of the distribution $\gamma_W(c)$. Nonetheless, it is often sufficient to use 20 % of the samples collected in a bin to achieve a reasonable estimate of C_U. In fact, with the smallest bin size and as few as 20 % of the samples $\varepsilon_C < 0.2$, which means the actual capacity should not be larger than 120 % of the estimated value. Another strength of this measurement technique is that errors are one sided (the actual value can only be larger than the measured), thus this measurement can be safely used as lower bound in resource optimization problems.

In addition, t_T must be taken slightly longer than the TTI to avoid the measurement being impacted by many bursty arrivals. In line with Eq. (1) of Section IV, $\Delta(t_T)$ approaches zero for $t_T > 15$ ms for most of the recorded flows.

Fig. 9, shows the normalized RMSE for various combinations of t_T and f. The bin size is set to 200 ms to give an example of this technique's results when it collects very frequent measurements. As expected ε_C decreases when t_T and f increase. For values of $t_T \geq 15$ ms and $f \geq 20$ %, the error is small enough for the model to give trustworthy results ($\varepsilon_C \leq 15$ %).

Finally, Table II shows some of the overall evaluation of the traces obtained with $f = 25$ % averaging over the bin size and using the optimal t_T ($\min t_T | \Delta(t_T) \to 0$). Optimal t_T and average asymptotic dispersion rate are computed mimicking the Speedtest method [12]. While some of the flows are transmitted using 2G EDGE data, the results are not included since there are too few such flows for statistical significance.

Technology	UMTS	HSPA	HSPA+	LTE
ADR (Mbps)	10.83	1.4	10.74	24.3
Optimal t_T (ms)	19	23	17	16

TABLE II: Average asymptotic dispersion rate and average optimal t_T per technology.

VII. Conclusions

We presented a lightweight measurement technique that leverages adaptive filtering over the packet dispersion time. This allows to estimate the available bandwidth in mobile cellular networks. Accurate estimates can be achieved exploiting as few as 5 % of the information obtained from TCP data flows. Given that this solution can support dense throughput sampling, it is ideal for bandwidth prediction and optimized resource allocation. In fact, if the future bandwidth availability is known, it is possible to predict when it is best to communicate by doing so when it is cheaper (i.e., more bandwidth available). In addition, our solution is able to estimate the fast capacity variations from a mobile terminal by monitoring the traffic generated under normal daily usage.

We validated our technique over a week-long measurement campaign spanning different locations, devices and communication technologies. We achieved good estimation accuracy even when using only short lived TCP connections. Since our technique is based on simple post-processing operations on the packet timestamps, it is possible to easily integrate it in background processes or OS routines.

We are planning to extend our measurement application with filter based prediction capabilities in order to provide mobile phones with a complete bandwidth forecasting tool, which, in turn, will allow for advanced resource allocation mechanisms. Finally, we are planning additional measurement campaigns in order further extend these encouraging results on passive and lightweight measurement tools.

Acknowledgments

The research leading to these results was partly funded by the European Union under the project eCOUSIN (EU-FP7-318398), SmartenIT (EU-FP7-317846), the Spanish MICINN grant TEC2011-29688-C02-01 and the German DFG (CRC 1053, MAKI).

References

[1] H. Abou-zeid, H. Hassanein, and S. Valentin. Energy-efficient adaptive video transmission: Exploiting rate predictions in wireless networks. *IEEE Transactions on Vehicular Technology*, 63(5):2013–2026, June 2014.

[2] N. Bui, F. Michelinakis, and J. Widmer. A model for throughput prediction for mobile users. In *European Wireless*, Barcelona, Spain, May 2014.

[3] N. Bui and J. Widmer. Mobile network resource optimization under imperfect prediction. In *Proc. IEEE WoWMoM*, Boston, MA, USA, June 2015.

[4] C. Dovrolis, P. Ramanathan, and D. Moore. Packet-dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Transactions on Networking*, 12(6):963–977, December 2004.

[5] A. Gerber, J. Pang, O. Spatscheck, and S. Venkataraman. Speed testing without speed tests: estimating achievable download speed from passive measurements. In *ACM IMC*, pages 424–430, Melbourne, Australia, November 2010.

[6] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *ACM MobiSys*, pages 225–238, Low Wood Bay, Lake District, United Kingdom, June 2012.

[7] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An in-depth study of LTE: Effect of network protocol and application behavior on performance. In *ACM SIGCOMM*, pages 363–374, Hong Kong, China, August 2013.

[8] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Sanadidi. CapProbe: a simple and accurate capacity estimation technique. *ACM SIGCOMM Computer Communication Review*, 34(4):67–78, October 2004.

[9] R. Kwan, C. Leung, and J. Zhang. Proportional fair multiuser scheduling in LTE. *Signal Processing Letters, IEEE*, 16(6):461–464, 2009.

[10] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '00, pages 283–294, New York, NY, USA, 2000. ACM.

[11] Z. Lu and G. de Veciana. Optimizing stored video delivery for mobile networks: The value of knowing the future. In *IEEE INFOCOM*, pages 2706–2714, Turin, Italy, April 2013.

[12] Ookla. Ookla speedtest mobile apps. http://www.speedtest.net/mobile/, last accessed June 2014.

[13] Y. Qiao, J. Skicewicz, and P. Dinda. An empirical study of the multiscale predictability of network traffic. In *Proceedings IEEE HDCP*, 2004.

[14] F. Ricciato, F. Vacirca, and M. Karner. Bottleneck Detection in UMTS via TCP Passive Monitoring: A Real Case. In *ACM CoNEXT*, pages 211–219, Toulouse, France, October 2005.

[15] N. Sadek and A. Khotanzad. Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model. In *Proceedings IEEE ICC*, 2004.

[16] R. L. Smith. Extreme value theory. *Handbook of applicable mathematics*, 7:437–471, 1990.

[17] S. Wang, Y. Xin, S. Chen, W. Zhang, and C. Wang. Enhancing spectral efficiency for lte-advanced and beyond cellular networks [guest editorial]. *IEEE Wireless Communications*, 21(2):8–9, April 2014.

[18] Q. Xu, S. Mehrotra, Z. Mao, and J. Li. PROTEUS: network performance forecast for real-time, interactive mobile applications. In *ACM MobiSys*, pages 347–360, Taipei, Taiwan, June 2013.

[19] Y. Xu, Z. Wang, W. K. Leong, and B. Leong. An end-to-end measurement study of modern cellular data networks. In *Passive and Active Measurement*, pages 34–45. Springer, 2014.