Bandwidth Estimation in Mobile Networks by Busy Period Detection

Zoltán Móczár, Sándor Molnár
High Speed Networks Laboratory, Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics, Budapest, Hungary
E-mail: {moczar, molnar}@tmit.bme.hu

***Abstract*—In the recent years, a significant research effort has been devoted to the development of bandwidth estimation techniques and tools due to the broad range of possible applications. The vast majority of bandwidth estimation algorithms are designed and optimized for wired networks. Therefore, these solutions not only provide inaccurate results in wireless environments but also rely on some information usually not known in advance or produce a severe additional load on the network. Specifically, in mobile networks the continuously varying characteristics of radio links make it an extreme challenge to estimate the currently available bandwidth. In this paper we present a bandwidth estimation method worked out for mobile networks, which models the dynamics of the bottleneck queue and identifies its busy periods. Our algorithm can estimate the unused bandwidth by exploiting the user-generated downlink network traffic with negligible extra load. The operation of the algorithm is demonstrated on real traffic traces captured by a mobile device in a 3G network.**

I. Introduction

Bandwidth estimation has received considerable attention in the last decades due to its key role in many areas of networking such as transport layer protocols, admission control, network management and multimedia streaming, just to mention a few. For example, transport protocols like TCP (Transmission Control Protocol) can use available bandwidth information to properly adjust the transmission rate, making possible the efficient utilization of network resources without causing congestion. Bandwidth estimation results also help network operators to identify the change of user demands by monitoring the network utilization and to plan capacity upgrades.

In this paper we present a bandwidth estimation algorithm for mobile networks with the following capabilities:

- no specific information about the network is needed (e.g. bottleneck link capacity);
- the bandwidth estimation algorithm runs only when the user is active (e.g. browsing the web), and a probe traffic is injected into the user-generated downlink network traffic;
- since the probe traffic consists of a sequence of small-sized packets, the estimation scheme causes a very low additional network load;
- by modeling the dynamics of the bottleneck queue and identifying the busy periods it can provide reasonable accuracy in spite of quick and high variations often seen in mobile data networks.

The paper is structured as follows. First, in Section II we discuss the notion of bandwidth together with its most important interpretations, and give a survey about the different models and tools worked out to estimate these measures. Section III introduces our bandwidth estimation scheme designed for mobile networks with detailed description. The operation of the algorithm is demonstrated in Section IV on real traffic traces captured in a 3G mobile network. Finally, Section V concludes the paper.

II. Related Work

Traditionally, bandwidth is used as a measure quantifying the data transfer rate that a network link or path can provide. However, it is important to distinguish between the different meanings of the term *bandwidth*. In the literature there are three frequently used interpretations: the *maximum possible bandwidth* that a link or path can deliver (capacity), the *maximum unused bandwidth* at a link or path (available bandwidth), and the *maximum throughput* can be obtained by a single TCP connection (bulk transfer capacity).

In the last decade, a plenty of bandwidth estimation algorithms and tools have been developed in order to meet the increasing demands [1], [2]. The design of efficient bandwidth estimation methods is not easy because some contradicting requirements are need to be fulfilled. An ideal algorithm would provide high estimation accuracy, fast operation and low overhead. However, in practice not all these features are equally relevant, and it highly depends on the application area which ones have to be optimized. For example, in case of transport protocols low overhead and low estimation time are required, but they do not need high accuracy for proper operation, a rough estimate is acceptable [2].

Available bandwidth estimation (ABwE) is one of the most challenging tasks in the context of bandwidth estimation methods addressed in many papers [1], [3]. The majority of ABwE techniques send probe packets to the receiver utilized in the estimation process and are based on two basic models: the Probe Gap Model (PGM) and the Probe Rate Model (PRM). PGM exploits the information about gap dispersion between two consecutive probe packets at the receiver. The gap dispersion has a strong correlation with the amount of cross-traffic in the tight link, that is, with the link having the lowest available bandwidth. The methods using PGM (e.g. Abing, IGI, Spruce) first determine the amount of cross-traffic, and then subtract the result from the known capacity of the tight link. PRM tools (e.g. Pathload, pathChirp, DietTopp) are based on the idea of self-induced congestion where probe packets

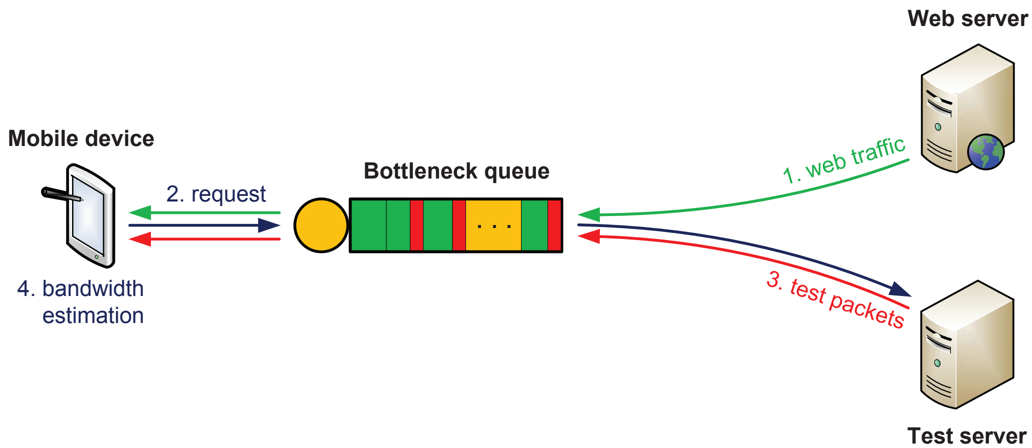


Fig. 1. Operation of the bandwidth estimation scheme

are sent at increasing rates to the receiver, and the available bandwidth is determined by studying the change of the queuing delay and measuring the output rate.

The issue of estimating the bulk transfer capacity (BTC) is investigated only in a few papers. BTC is defined as the maximum throughput can be obtained by a single TCP connection [4]. For example, Allman introduces a BTC measurement tool in [5] and presents its empirical evaluation together with the investigation of reliability. Gardner et al. propose a novel method for estimating the BTC of an IPv6 network path, conducted from a single point to a non-instrumented target [6]. In fact, BTC is very hard to measure since it can be affected by several factors such as the type of cross-traffic, the number of competing TCP connections, the buffer space in routers along the network path and queuing policies [1].

Unfortunately, the vast majority of bandwidth estimation tools discussed above are designed for wired networks, therefore they cannot provide accurate results and short convergence time in wireless environments, especially in mobile networks. Cellular networks bring a lot of additional challenges, which make bandwidth estimation more difficult compared to wired networks. The bandwidth available by the user is continuously varying due to the changing network conditions such as the location and motion speed of the mobile device, the number of users in the current cell, the signal strength, handovers and many other effects [7]. Negreira et al. discuss the issues of end-to-end measurements over GPRS-EDGE networks in [8] and present a new methodology capable of providing acceptable results in such environments. The authors of [9] defines a so-called in-context network performance measure to express the user experience when they are interacting with their mobile devices. They carried out a large-scale measurement study using data collected across cell subscribers and controlled experiments. They pointed out that, to obtain accurate results, performance measurements must be conducted on devices, which are actively used during the measurement time frame, currently exchanging limited user traffic and can be found in the same position and environment since the last usage of the device. Bergfeldt et al. performed an evaluation of bandwidth measurement tools over a high-speed downlink UMTS channel by running experiments in a commercial mobile network [10]. In this work they investigated several test scenarios by using various types of cross-traffic and bottlenecks. The results showed that algorithms can significantly under- or overestimate the available bandwidth under certain network conditions.

III. Bandwidth Estimation Algorithm for Mobile Networks

In this section we introduce our bandwidth estimation scheme designed for mobile networks. First, the main concept is presented by a high-level description, then the operating mechanism of the algorithm is given in detail.

A. Basic Idea

Our bandwidth estimation method is designed to estimate the *unused bandwidth* available on a mobile device by exploiting the user-generated downlink network traffic with negligible extra load. Figure 1 shows the main concept in an architectural view with the main hardware and software components.

The bandwidth estimation scheme works as follows. The downlink network traffic generated by the user is continuously monitored on the mobile device. When the type and amount of traffic are considered as sufficient to initiate the bandwidth estimation process, the mobile device sends a request signal to the test server. The test server starts to generate a sequence of test packets with a specified frequency and sends it towards the mobile device. The generation of test packets at the test server is finished if a stop request is received from the mobile device or a timeout is occurred. By this way test packets are injected into the user-generated downlink traffic. Our algorithm running on the mobile device simply estimates the unused bandwidth by dividing the amount of traffic observed between two test packets by the elapsed time. However, it can be highly inaccurate, therefore the key step is to determine which periods of the data flow are eligible for performing estimation. The algorithm utilizes two basic information to achieve this: (1) the fixed test packet generation interval and (2) the measured test packet inter-arrival times (IAT). Based on these information our method models the queue dynamics of the bottleneck link (assumed to be the wireless connection between the base station and the mobile device) and identify its busy periods in order to enhance the estimation accuracy.

B. Algorithm Description

The most challenging task is to capture the busy periods of the bottleneck queue. In other words, we find those intervals in the downlink traffic trace when the queue is not empty, and hence, enqueued packets will be serviced at maximal rate. The pseudocode of the algorithm can be seen in Algorithm 1 where d, th, and gap denote the delay between the generation of test packets, the positive threshold used for busy period detection and the highest IAT can be accepted in the queue modeling phase, respectively. Furthermore, t_i is the arrival time of the i^{th} test packet captured at the mobile device, t_{i+1} is the arrival time of the $(i+1)^{th}$ test packet and n is the number of test packets in the traffic trace. The boolean variables m and b indicate if the queue modeling and busy period detection phases are active.

Algorithm 1: Bandwidth estimation algorithm

input : $trace, d, th, gap$
output: bw
$m \leftarrow false$; $b \leftarrow false$;
for $i \leftarrow 1$ **to** $n-1$ **do**
    if $t_{i+1} - t_i = d$ **and** $m = false$ **then**
        $q \leftarrow 0$;
        $m \leftarrow true$;
    else if $t_{i+1} - t_i > gap$ **and** $m = true$ **then**
        $m \leftarrow false$;
        $b \leftarrow false$;
    else if $m = true$ **then**
        $q \leftarrow t_{i+1} - t_i - d + q$;
        if $q \geq th$ **and** $b = false$ **then**
            $s \leftarrow t_i$;
            $b \leftarrow true$;
        else if $q < th$ **and** $b = true$ **then**
            $rates \leftarrow \texttt{Add}\left(\frac{\text{amount of traffic in } [s,t_i]}{t_i - s}\right)$;
            $b \leftarrow false$;
        end
    end
end
return $bw \leftarrow \texttt{Mean}(rates)$;

In the followings we discuss each main step performed by our bandwidth estimation scheme:

- **Step 1 *(initialization).*** As a first step, the algorithm detects whether the queue is empty by finding an IAT of two successive test packet arrivals, which is equal to the test packet generation interval ($t_i - t_{i-1} = d$). In ideal case, it means that the i^{th} test packet will experience zero waiting time ($q_i = 0$).

- **Step 2 *(queue dynamics modeling).*** The algorithm starts to model the queue dynamics and computes the current waiting time by determining the difference between the IAT ($t_i - t_{i-1}$) and the generation interval (d). In the following time slots current waiting time comes from the sum of this difference and the waiting time calculated in the previous slot ($q_i = t_i - t_{i-1} - d + q_{i-1}$). In theory the result should be a non-negative value, but in practice there are some factors (e.g. jitter), which can turn it to negative.

- **Step 3 *(busy period detection).*** Ideally, when the cumulative waiting time (referred to as queue length) becomes greater than zero, we could say that the queue is busy. However, to make the detection more accurate in realistic environments, the algorithm uses a positive threshold (th) instead of zero as a reference point to identify the start of the busy period (see Figure 2). If the waiting time drops below this threshold, it indicates the end of the busy period. Another effect which can lead to the termination of the busy period detection and the queue modeling phase is observing a high test packet IAT probably not due to the impact of user-generated traffic. We call these IATs as outliers, and the outlier detection can also be controlled by a threshold parameter (gap). When the queue modeling phase ends, the algorithm finds the next empty state of the queue to restart the modeling process.

- **Step 4 *(bandwidth estimation).*** Once the busy periods are captured the algorithm performs bandwidth estimation by computing the fraction of the amount of downlink traffic observed between the first and last test packets of the busy period and the elapsed time. We note that, depending on the length of the traffic sample, our method may identify several busy periods. The final result will be the mean of the estimated values.

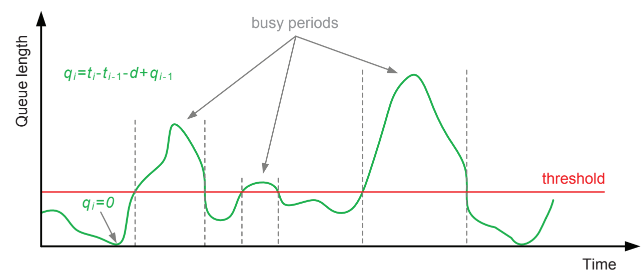


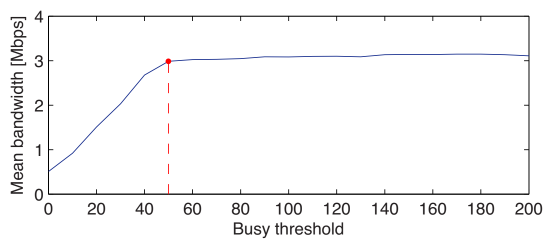
Fig. 2. Busy period detection by modeling the queue dynamics

IV. Evaluation Results

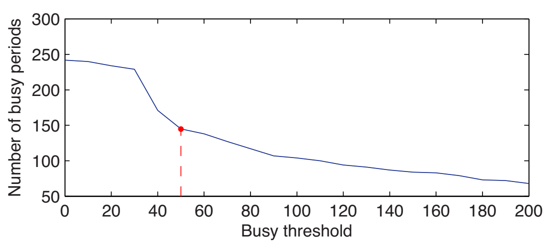
To evaluate the operation of our heuristic bandwidth estimation scheme we conducted several measurement scenarios in a controlled environment. In this section we show an example from our test results obtained on a real traffic trace captured in a 3G mobile network. As discussed in Section III the algorithm was designed to estimate the currently available bandwidth by exploiting the user-generated downlink traffic. To examine typical user behaviors, we used a multi-functional network traffic emulator presented in [11]. This tool can accurately simulate different types of user activity such as web browsing, or the use of video streaming services (e.g. YouTube) and social networking applications (e.g. Facebook).

The measurements were performed on a smartphone with HSDPA support and Android operating system where we generated realistic web traffic based on user behavior emulation. The test packets were sent periodically from the server towards the mobile device over UDP with an inter-arrival time of 100 ms. We measured the IAT distribution at the receiver with no cross-traffic and observed that the time spaces between consecutive UDP packets were only slightly changed (in the order of few milliseconds). However, to take this effect into account, mostly caused by the variation of signal quality, a positive busy threshold was applied in the queue modeling phase according to Algorithm 1. In order to mitigate the network load induced by the test traffic, small-sized (i.e. 60 bytes) UDP packets were injected into the user's downlink stream. At the mobile device we captured 60 minutes long packet traces for evaluation purposes and carried out an extensive analysis by investigating many different aspects.

The following results present the traffic intensity for the measured and estimated time series, the busy period statistics, as well as the histograms and distribution functions of the download rate. To identify the busy periods of the bottleneck queue we used a positive threshold of 50, and for outlier detection we defined the maximum acceptable inter-arrival time as 1000 ms. During our evaluation tests we experimented with different busy thresholds and concluded that a positive value has to be applied in order to filter out the impact of some undesirable phenomena like jitter. However, in general, above a certain threshold we get very similar estimation results including the distribution and mean of the estimated bandwidth values (Figure 3a), but a higher value leads to a smaller number of detected busy periods over a given time interval (Figure 3b). To obtain the best outcome it is practical to choose the lowest possible threshold which otherwise can be considered as sufficient to avoid the issues mentioned above.



(a) Mean estimated available bandwidth



(b) Number of busy period samples

Fig. 3. The choice of busy threshold

Figure 4 shows the measured downlink traffic intensity in one second resolution and the estimated available bandwidth calculated for the busy periods. Web traffic is eligible to demonstrate the capabilities of our bandwidth estimation method since typical users frequently check their emails and favorite social networking sites, or simply browse the web. Our main goal was to design such an algorithm, which can give an estimate for the unused bandwidth even if the user generates only a small amount of network traffic, for example, by web browsing. The figure indicates that the downlink traffic highly fluctuates due to the characteristics of user activity, but we can identify many intervals when a page load utilizes the instantaneous available bandwidth. This means that during several periods of time the bottleneck queue is busy, or in other words, it works at the maximum service rate. The figure depicts the estimated bandwidth calculated for these busy periods. We emphasize that it is really hard to give an accurate estimation, because in a mobile network available bandwidth is continuously changing and affected by many conditions like motion speed, the number of users in the current cell, signal strength, handovers, and so on [7], [8]. In spite of this fact, one can see that the busy periods identified by our heuristic algorithm covers well the highest download rates offered by the network.

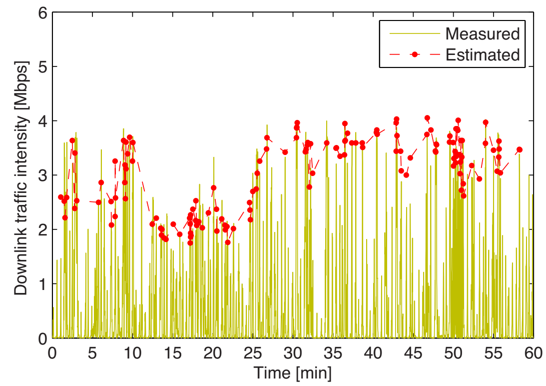


Fig. 4. Traffic intensity

Figure 5 presents the relative frequencies of busy period lengths. The results clearly show that busy periods are quite short in case of web traffic. Specifically, almost 50% and 75% of all captured busy periods are shorter than half and one second, respectively. Web browsing typically results in bursty traffic since users spend at least a few seconds on a page before proceeding. Nevertheless, the length of the downloading periods is still sufficient to calculate proper bandwidth estimates.

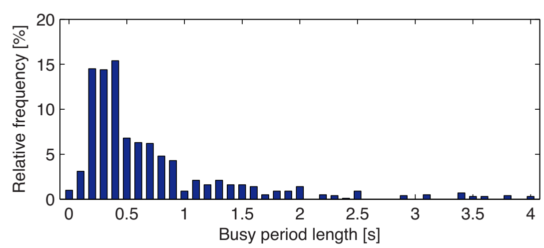
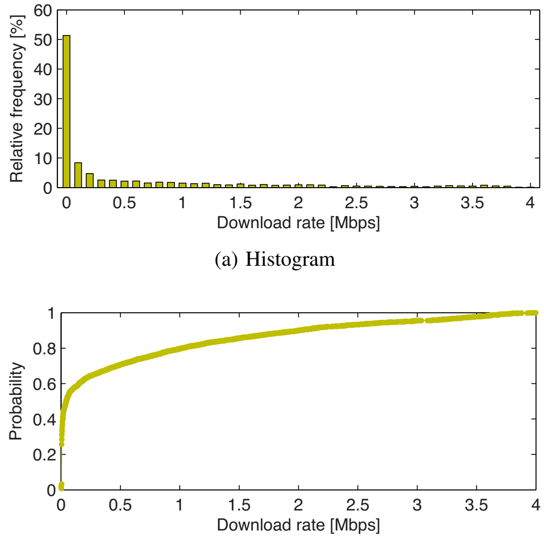


Fig. 5. Busy period statistics

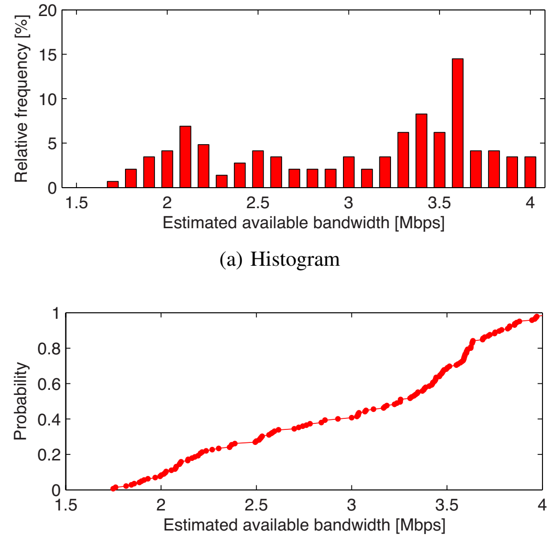
Figure 6 and Figure 7 depict the histogram and the cumulative distribution function of the measured download rate and the estimated available bandwidth, respectively. Looking at Figure 6 we can find low download rates much more frequent compared to high rates in the measured packet trace. This is due to the phenomenon discussed earlier in the paper, namely, the maximum bandwidth is utilized only in the cases of traffic bursts, which are separated by idle periods. For example, more than 70% of transmission rates fall below 0.5 Mbps, because once a page is loaded no further network traffic is usually generated or only small amount of data is need to be



(a) Histogram

(b) CDF

Fig. 6. Measured rate characteristics



(a) Histogram

(b) CDF

Fig. 7. Estimated bandwidth characteristics

exchanged (e.g. for online advertisements). Our purpose was to capture those intervals when downloading consumes the available bandwidth. We ran 100 rounds of the widely used *Speedtest* [12] on the mobile device with half minute breaks before and after the one hour long measurement period. The perceived available downlink bandwidth was between 1.6 and 4.2 Mbps in accordance with our estimation results calculated for the busy periods, see Figure 7a. Furthermore, the mean bandwidth provided by Speedtest was 3.1 Mbps, which is also very close to our estimate of 3 Mbps (Figure 3a). As pointed out in the discussion of Figure 5 busy periods are short in time. Moreover, Figure 7b suggests that web traffic originated from a typical smartphone user contains small number of busy periods, hence it is crucial how accurately the detection method can capture them. While each estimated bandwidth value exceeds 1.7 Mbps, about 85% of measured rates are below this limit (Figure 6b), accordingly, do not fall into any of the identified busy periods.

V. Conclusion

Bandwidth estimation in cellular networks is challenging due to the nature of radio communication. Currently available bandwidth is a continuously changing metric affected by numerous environmental factors. In this paper we proposed a heuristic approach, which can exploit the user-generated traffic and is capable of modeling the dynamics of the bottleneck queue and capturing its busy periods. We demonstrated the operability of our algorithm on a packet trace gathered in a 3G mobile network by using a realistic traffic emulator. It has been found that busy periods can be relatively short, but the presented method is able to capture them with high reliability. The results suggest that proper identification of busy periods makes it possible to estimate the available bandwidth.

Acknowledgment

This research work was supported by Ericsson Hungary and the High Speed Networks Laboratory (HSNLab) at Budapest University of Technology and Economics. The authors gratefully acknowledge the kind help of László Kovács, András Veres and Tamás Borsos from Ericsson TrafficLab.

References

[1] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy, “Bandwidth Estimation: Metrics, Measurement, and Tools”, *IEEE Network*, vol. 17, pp. 27–35, 2003.

[2] C. D. Guerrero, M. A. Labrador, “On the Applicability of Available Bandwidth Estimation Techniques and Tools”, *Computer Communications, Elsevier*, vol. 33, no. 1, pp. 11–22, 2010.

[3] J. Strauss, D. Katabi, F. Kaashoek, “A Measurement Study of Available Bandwidth Estimation Tools”, *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pp. 39–44, Karlsruhe, Germany, 2003.

[4] M. Mathis, M. Allman, “A Framework for Defining Empirical Bulk Transfer Capacity Metrics”, *RFC 3148, IETF*, 2001.

[5] M. Allman, “Measuring End-to-End Bulk Transfer Capacity”, *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pp. 139–143, San Francisco, CA, USA, 2001.

[6] R. Gardner, F. Garcia, “Bulk Transfer Capacity Estimation in IPv6 Networks”, *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*, pp. 1–6, Bucharest, Romania, 2006.

[7] U. Hentschel, A. Schmidt, A. Polze, “Predictable Communication for Mobile Systems”, *Proceedings of the 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 24–28, Newport Beach, CA, USA, 2011.

[8] J. A. Negreira, J. Pereira, S. Pérez, P. Belzarena, “End-to-End Measurements Over GPRS-EDGE Networks”, *Proceedings of the 4th IFIP/ACM International Latin American Conference on Networking*, pp. 121–131, San José, Costa Rica, 2007.

[9] A. Gember, A. Akella, J. Pang, A. Varshavsky, R. Caceres, “Obtaining In-Context Measurements of Cellular Network Performance”, *Proceedings of the 12th ACM International Conference on Internet Measurement*, pp. 287–300, Boston, MA, USA, 2012.

[10] E. Bergfeldt, S. Ekelin, J. M. Karlsson, “A Performance Study of Bandwidth Measurement Tools over Mobile Connections”, *Proceedings of the 69th IEEE International Vehicular Technology Conference*, pp. 1–5, Barcelona, Spain, 2009.

[11] S. Molnár, P. Megyesi, G. Szabó, “Multi-Functional Emulator for Traffic Analysis”, *Proceedings of the 2013 IEEE International Conference on Communications*, pp. 2397–2402, Budapest, Hungary, 2013.

[12] Ookla Speedtest, http://www.speedtest.net/