LTE or WiFi? Client-side Internet link selection for smartphones

Paweł Foremski and Krzysztof Grochla

The Institute of Theoretical and Applied Informatics
of the Polish Academy of Sciences
Bałtycka 5, 44-100 Gliwice, Poland
{pjf, kgrochla}@iitis.pl

Abstract. Current mobile phones and tablets are equipped with two technologies for accessing the Internet: WiFi and Cellular. Deciding which of these two interfaces provides faster data transfer is often non-trivial, but most of the currently used devices use a simple priority scheme that prefers WiFi to Cellular.
In this paper, we propose a novel system that automatically selects the best link available in the current user location. The system periodically probes the bandwidth available on both links and makes statistical predictions, while avoiding excessive data and battery usage. We experimentally validated our approach using a dedicated application for Android.

Key words: Cognitive Networks, Available Bandwidth Estimation, Cellular Networks, LTE, SON, Mobility Management, Offloading

1 Introduction

Current mobile phones and tablets are equipped with multiple wireless interfaces. Apart from using the UMTS or LTE technology, they transmit data using WiFi in areas where it is available. The choice of the wireless interface is typically static. The Android and iOS systems employ a priority-based selection scheme, which chooses WiFi whenever possible, or the Cellular link instead. However, this simple policy is inefficient in terms of energy use and achieved transmission speed: it requires both interfaces to be enabled virtually all of the time, which decreases battery lifetime, and there are many situations in which WiFi is slower than Cellular, e.g. in heavily loaded public WiFi networks and in locations connected over a low-speed ADSL line.

Some users manually disable the WiFi interface for most of their time, and manually enable it only when in range of a known AP. Such solution increases battery lifetime, but requires manual control and takes time. To automate this process, a few tools were proposed—e.g. Sony location-based WiFi or Smart WiFi Toggler [1]—but they do not evaluate the available Internet bandwidth.

In this paper, we propose a method to automatically select the optimal interface in terms of maximum transmission speed, with minimal energy usage and

Preprint version - final paper to appear in:
Foremski P., Grochla K., "LTE or WiFi? Client-side Internet link selection for smartphones", Communications in Computer and Information Science. Proceedings of the 22nd International Conference on Computer Networks, CN2015. Springer, 2015
(Available at link.springer.com)

data transfer. We introduce a lightweight tool for estimating available bandwidth of WiFi and Cellular links, and we present a novel algorithm to select the best interface in the current user location.

The rest of the paper is organized as follows: in Section 2 we present the background work and the motivation for the proposed solution, in Section 3 we describe the proposed methodology to select optimal interface for data transmission. Section 4 covers the experimental validation of the method using a smartphone application and analysis of the stability of the link selection. We finish the paper with a short conclusion in Section 5.

2 Background

The rapid growth in the number of wireless devices used for accessing the Internet substantially increased the traffic load on mobile networks. The global mobile traffic grew by 81% in 2013 [2], which results in high load on the currently deployed wireless networks, and—as a consequence—to decreased quality of service in some locations during peak hours. These changes make it hard for the user to manually select the optimal Internet connection.

Most of the currently produced smartphones are equipped with two radio interfaces: 3G/LTE (Cellular) and IEEE 802.11 (WiFi) [3]. These two types use different frequencies and media access methods: while WiFi is based on random channel access, the Cellular networks channel access is managed by the base station. WiFi operates in ISM band, in which anyone can easily start transmission, while Cellular networks use licensed bands, in which transmission is controlled by network operator. In both technologies the throughput of the transmission is limited by the radio signal propagation conditions, radio bandwidth available, and the amount of devices sharing the same radio resources.

The two most widely used operating systems for smartphones, Android and iOS, by default use WiFi if both connections are available. According to [3], 2/3 of consumers prefer WiFi to Cellular. WiFi is free in most cases, while the Cellular data plan requires a monthly fee. WiFi is also often perceived as more efficient than Cellular. While the IEEE 802.11 standard offers very high transmission rates of up to 300Mbps in local networks, the actual bandwidth is often limited by an ADSL link to which the WiFi access point is connected, e.g. between 2 and 25Mbps. In crowded locations, where many users share the same backhaul connection or where many interfering APs are deployed, the WiFi performance is heavily degraded [4]. On the other hand, the average throughput offered by LTE networks varies between 9Mbps [5] and 13Mbps [6], which is higher than the throughput of a low-cost ADSL link. The performance of an LTE connection depends on the distance to the base station and on the number of users transmitting data through it, so it can significantly change in space and time. The measurement presented in [5] shows that it may change between 0.6Mbps per 5 percentile to 24Mbps for the 95 percentile. Thus, deciding whether WiFi or LTE offers a faster transmission is not an easy task for the user.

The problem of transferring the data traffic from Cellular to WiFi was heavily investigated in the literature, but most of these works evaluated it from the network operator perspective, which aims at offloading transmissions towards the unlicensed bands [7]. The 3GPP Release 10 defined data offloading as a key solution to cope with the constantly increasing load on packet data networks [8]. Offloading to WiFi is considered jointly with small cell deployments [9]. However, implementation of the infrastructure to manage the offloading from the operator perspective is costly and requires economical relations between the network operator and the owners of access points [10].

From the client perspective, users want to simply use the interface that offers the fastest data transfer in their current location. Users may manually enable or disable the WiFi interface, but this consumes time and introduces burden. There are applications that automate this process and disable the WiFi interface in locations where the user configured the phone to do so, e.g. Smart WiFi Toggler [1]. However, the selection of the best link in specific location should estimate the available bandwidth on both WiFi and LTE interfaces, and realize the selection quickly and in an energy efficient way. This can be achieved by measuring on the client side which of the two interfaces provides faster access to the Internet. The throughput estimation should minimize the amount of data transferred, to minimize the cost of its use and minimize the energy utilization. To the best of our knowledge, there is no such tool currently available.

3 Selecting the best link automatically

In this Section, we describe our method for selecting the best Internet link on mobile devices. The basic idea is to make periodic and lightweight measurements of the instantaneous download speeds in locations where the user has active WiFi connection. We summarize these measurements using statistics and select Cellular link if it performs better than WiFi in the current location.

3.1 Available Bandwidth Estimation

Available end-to-end bandwidth is an important metric of an Internet path, which has high impact on the quality of an Internet link in general. Numerous methods for measuring this metric were proposed in the literature, under the name of Available Bandwidth Estimation (ABE) [11–13]. In [14], the authors experimentally compared 9 ABE tools in the same networking environment, in terms of intrusiveness, response time, and accuracy in presence of different cross-traffic streams. However, only Spruce [12], pathChirp [11], and Assolo [13] generated less than 500KB of traffic per measurement. We did not consider the other methods because mobile operators put monthly limits on Cellular data transfer. We chose Assolo as a state-of-the-art method, because it is an optimized version of pathChirp, and has lower intrusiveness than Spruce.

However, in [15] the authors showed that current ABE tools will not work in large-scale distributed systems. The authors reported a significant under-estimation of the available bandwidth, with divergence of estimations vs. real values. In our paper, we experimentally confirm these results in Section 4: Assolo cannot reliably estimate the bandwidth of an artificially limited Internet link (see Fig. 2). Thus, we propose a new lightweight ABE tool: pik.

The basic idea behind pik is to send a short peak of UDP data and measure its duration at the receiver. We also estimate the Round-Trip Time (RTT) for better accuracy and to work-around various network buffers. The measurement process is as follows. The client, which is the receiver part wanting to know its available bandwidth, registers at the server and obtains a random password used for further authentication. Next, the client sends a `PING` request, to which the server replies with a 50B response. This step is repeated by default 5 times with a 1-second timeout, and the average time between sending the request and receiving the response is treated as the link RTT. Finally, the client sends a `START` request, to which the server replies with a peak of data, by default 100 packets of 1KB length. The server sends the data to the network as fast as possible, in a single loop without any pauses. The client assumes reception of the first packet at the time of the `START` request plus RTT, but the time of the last packet is measured. The duration of the peak at the receiver side is calculated, and finally the available bandwidth is the amount of data received divided by the peak duration. This final stage has a time limit of 3 seconds by default.

The pik tool works well for Internet links with artificial limits, i.e. bandwidth caps set by an ISP operator, but we need to repeat the measurement several times to gain reliable information on the link performance.

3.2 Link selection

Basing on experimental evaluation, we propose the following condition to select the Cellular connection instead of WiFi:

$$0.75 \cdot \overline{M_c^{(L)}} > \overline{M_w^{(L)}}, \tag{1}$$

$$M_c^{(L)} = \{c_1^{(L)}, \ldots, c_n^{(L)}\} \quad n \geq 5, \tag{2}$$

$$M_w^{(L)} = \{w_1^{(L)}, \ldots, w_m^{(L)}\} \quad m \geq 5, \tag{3}$$

$$L = (\text{SSID},\ \text{BSSID}), \tag{4}$$

where $M_c^{(L)}$ is a set of pik measurements for Cellular at location L, $M_w^{(L)}$ is the same for WiFi, and L is a tuple of SSID and BSSID for the associated WiFi AP.

The goal of Eq. 1 is to select Cellular if on average it performs much better than WiFi in the current location, e.g. if it is a few Mbps faster. We highlight that the goal of our work is not estimating the available bandwidth, but choosing the better performing link on mobile devices. The bandwidth available to the user depends on many factors, e.g. the number of contending hosts in a WLAN network or on the scheduling algorithm in LTE, which is dynamic. Thus, we propose to periodically repeat the measurements several times and make the decision using link statistics. Hence, we present a heuristic approach validated through experiments, instead of comparing the bandwidths directly.

We propose to update $M_c^{(L)}$ and $M_w^{(L)}$ periodically in the background, without disrupting normal operation of the device. A link selection system can schedule bandwidth measurements each few hours, provided that the screen is off and a configured WiFi network is available. New data should replace old measurements. However, there is a trade-off between frequent updates and the amount of transferred data, hence the update rate should be chosen wisely. Finally, we believe that updating $M_c^{(L)}$ and $M_w^{(L)}$ without active measurements, e.g. by observing the interface byte counters, is prone to errors. One could not assume that current transmissions are not band-limited at the sender side (e.g. video streams). Mobile devices allow for only one active connection, so it would be cumbersome to passively profile two links at the same time.

3.3 Practical application

We implemented a practical link selection system for the Android platform as "BX Network" application[1]. The application has two operation modes: 1) *Active*, when the device screen is on and unlocked, and 2) *Sync*, when the screen is off. Basically, BX Network runs the link selection algorithm (Eq. 1) when entering the Active mode and uses the Android API to apply the results. When leaving this mode, all links are switched off. However, while in the Sync mode, the application periodically enables both links for a short period of time, letting the Android synchronization to run. In such cases, BX Network also collects pik measurements if possible and desired.

We implemented pik for Android using native API (NDK), for performance reasons. Measurements are governed by a scheduler that runs pik at most every 3 hours in the same location. If there are less than 5 measurements in the last 7 days for the current location, the scheduler allows more frequent updates, to collect the data for Eq. 1 as fast as possible. On the other hand, if no configured WiFi network is available, no pik measurements are made. Results are stored in a SQLITE database. Whenever the system needs to select the best link, it fetches location information from the Android WiFi API, and queries the database for pik results in the current user location, for the last 7 days.

The final effect of running BX Network on a smartphone is that it automatically switches to the best available Internet connection in a few seconds after unlocking the screen. The delay is due to the WiFi scanning procedure implemented in recent versions of Android.

4 Experimental validation

We base our experimental validation on 3 data sources: `DS1`) bandwidth tests to two distant hosts based in US (New York) and Poland (Poznan), repeated 100 times in 24 hours, `DS2`) bandwidth tests over an artificially limited link to the host based in US, repeated 480 times during 3 hours, and `DS3`) measurements

[1] See `https://play.google.com/store/apps/details?id=com.bxlabs.network`

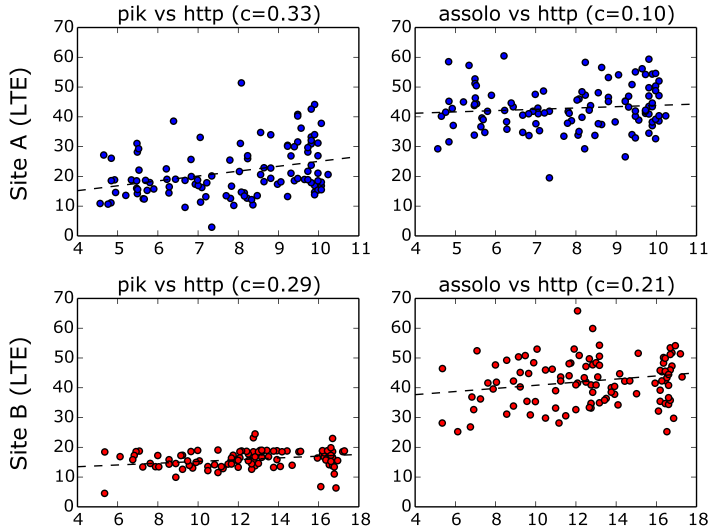

Fig. 1. HTTP download speed (horizontal axes, Mbps) versus speed estimated by pik and Assolo (vertical axes, Mbps). Measurements repeated 100 times for 24h using LTE. Correlation coefficients shown in c, least-squares fit shown as dashed lines.

collected during typical usage of BX Network on a single smartphone for 40 days. For DS1 and DS2 we used three idle and stationary Internet uplinks—LTE, WiFi, and Ethernet—of which the Ethernet link was the fastest. All data was collected during Nov 2014-Jan 2015. We conducted 4 experiments described below.

Experiment 1: Estimating link speed using pik and Assolo. In Fig. 1, we use DS1 to compare ABE tools against real bandwidth attained while downloading a 3MB file from a web server, over an LTE link. We tested if a single run of a lightweight ABE method can estimate the real speed available to user while downloading a medium-sized web object. We repeated the experiment 100 times, running pik, Assolo, and an HTTP client (wget) immediately one after another. We did not limit the server link nor the LTE link in any way: we assume that the bandwidth changes were due to network congestion and load on the LTE base station. The results show that neither pik nor a state-of-the-art tool can estimate the HTTP download speed reliably. We see some correlation, but the results are generally random, especially for Assolo. Both methods over-estimate the HTTP speed by a factor of 1.5-5 (pik) or 4-12 (Assolo). However, pik results are generally more stable and closer to reality.

In Fig. 2, we show situation in which the link is artificially limited on the server side using Token Bucket Filter (TBF), with rates increasing from 1 till 16Mbps (DS2). Again, we compare the HTTP download speeds with the values estimated by pik and Assolo. The results show almost perfect estimation using pik (correlation equal to 1.0) and mediocre results using Assolo (correlation equal to 0.45). The pik method slightly under-estimated the HTTP speed, while Assolo

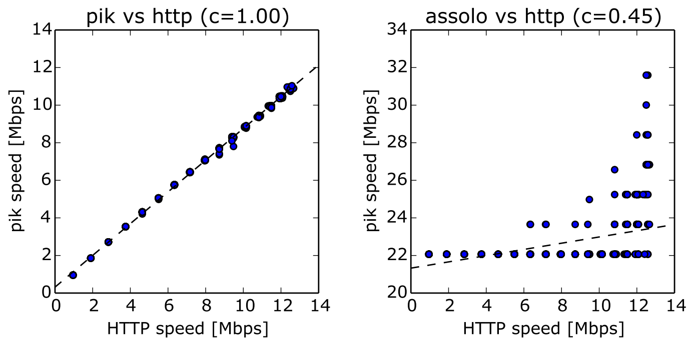

Fig. 2. HTTP speed vs estimates by pik and Assolo. Bandwidth limited using Token Bucket Filter (1-16Mbps). Measurements repeated 30 times over an Ethernet link.

over-estimated the real values by an order of magnitude (for small TBF rates). Thus, in case the link is limited by the network operator, pik reliably estimates the available bandwidth, while Assolo does not. We conclude *Experiment 1* that it is difficult to predict the HTTP download speed using small amount of data, but evaluated tools demonstrated some ability, of which pik was better and more reliable than Assolo. Thus, we can use pik for quick measurements of instantaneous link speeds.

Experiment 2: Selecting the best link with statistics. In Fig. 3, we search for the statistical method that would select the best link in `DS1` reliably. We take random samples of all pik measurements for 3 Internet links and we apply statistical measures of location: arithmetic mean and quartiles (Q1, Q2, and Q3). The link with the highest value gets selected as the best link. Whenever the algorithm selects Ethernet, we treat it as a success. We repeated the experiment 100 times for various sample sizes, presenting the average for two distant hosts. The results show that in most cases the bigger is the sample, the better. We obtained the best results by using the value of the average and the first quartile. We recommend using the arithmetic mean for its simplicity and popularity (it is available in SQLITE). The sample size should be at least 5.

In Fig. 4, we present results obtained from `DS2`: LTE and WiFi speeds measured using pik for increasing TBF rates. The upper plots show raw measurements, and the lower plot presents their statistics. Basing on the results presented in Fig. 3, we chose to apply the arithmetic mean of 15-element random samples. We see that the accuracy for LTE is generally better than for WiFi. The mean absolute error was 0.74Mbps for LTE and 2.3Mbps for WiFi, which is reflected in the fact that the estimated bandwidth for LTE much closely follows the TBF limit. Thus, we conclude that Eq. 1 can be used to choose LTE in favor of WiFi, but if it is much faster. By applying statistical analysis to raw pik measurements we obtain meaningful results.

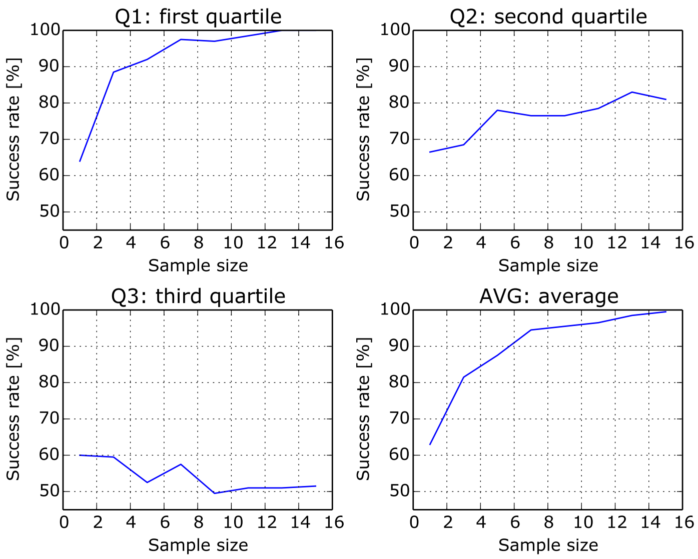

Fig. 3. Probability of choosing the best link for various statistics of pik measurements. Experiments repeated 100 times for each sample size.

Experiments 3 and 4: Stability and costs. In Fig. 5, we evaluate stability of our link selection method. Using data in DS3, we make a link selection decision basing on a random sample of all measurements. Then, we simulate the impact of a new measurement on the decision, by adding one more measurement to the sample and evaluating the arithmetic mean once again. If the decision does not change, we treat the algorithm as stable. Fig. 5 presents results obtained for two user locations, repeated 10,000 times for sample size ranging from 1 till 15. The results show the bigger is the sample size, the better. For given data source, the algorithm is stable in 99% of cases for 5-element random sample, and 100% for at least 10-element random sample.

	Typical user		Active user	
	Work days	Holidays	Work days	Holidays
Amount per year	230	135	265	100
Time at home [h]	12	16	10	12
Time at work [h]	8	0	9	0
Other activities [h]	4	8	5	12
Used WLANs per day	3	1	5	2
Measurements per day	7	5	10	7
Data usage per day [KB]	714	510	1020	714
Average per month [KB]	19,170		28,080	

Table 1. Rough simulation of Cellular data costs.

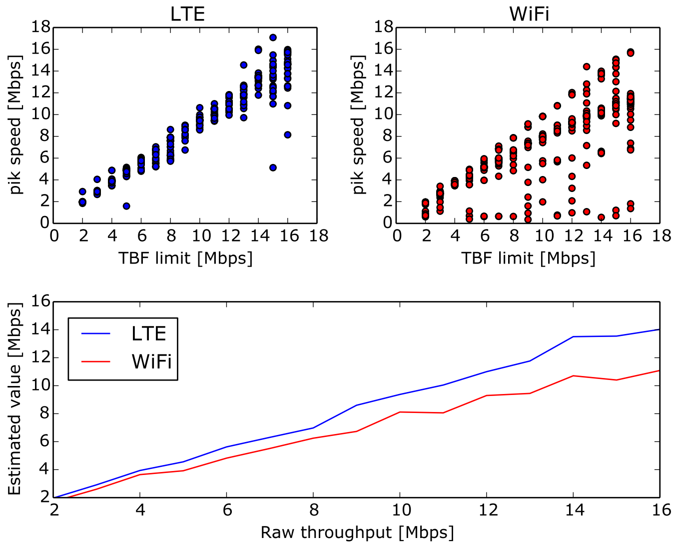

Fig. 4. Estimating TBF-limited link speed on LTE and WiFi. The upper plots show raw pik measurements, while the lower plot presents their statistics.

In our last experiment, we extracted real Cellular data usage from DS3, which should roughly illustrate the real monthly costs of using BX Network. The application was active for 40 days and used the scheduler described in Section 3 for measuring LTE and WiFi speeds on a typical smartphone. During the test period, the application made 220 Cellular measurements. For each measurement, pik used 100 data packets of 1000B and 10 ping packets of 50B, which corresponds to 102KB per measurement (including packet overhead). Thus, it used 561KB a day, or 16MB per month. In Tab. 1 we simulate monthly Cellular data costs for two scenarios: typical user and active user. We assume that the user has various network usage patterns for work days and weekends, and in different places [16]. Assuming the user accesses a WLAN at home, at work, and sometimes in travel, we propose various amounts of measurements per day. However, even for an active user, monthly Cellular data usage should stay below 30MB.

In this work we don't target the energy usage minimization as the goal of the optimization. Measurements in [6] show that transfer of the same amount of data on LTE requires 5.4 - 12 times more energy than on WiFi, so if we calculate only the energy used for transfer it is always more efficient to turn on WiFi when it is available. However the total energy usage also depends on the time spent during waiting for the data. This time is linearly proportional to the network bandwidth available for both LTE and WiFi. The amount of energy consumed by proposed application, given our assumptions on Cellular data costs (Table 1), is proportional to transferring a few MB per day. Existing mobile applications for testing Internet speed, like SpeedTest [17], transfer an order of magnitude more data per single measurement, which requires much

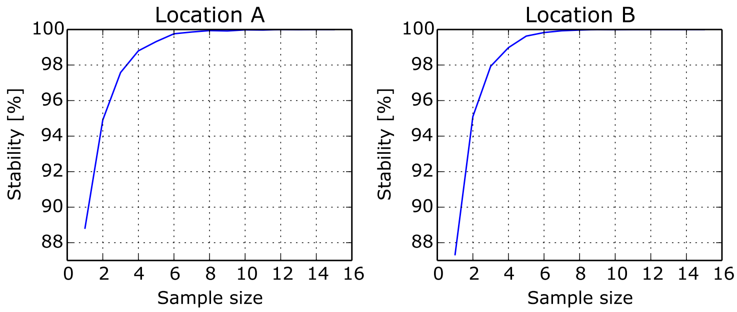

Fig. 5. Probability that a new pik measurement does not change the decision on the best link. Experiment repeated on real data 10,000 times for each sample size.

more energy. Our application also improves energy efficiency by switching off the wireless interfaces when they are not needed, reducing the energy usage in standby mode. In summary, the comparison of the total energy consumed with or without automatic link selection require more depth analysis, which we leave for further study.

5 Conclusions

In this paper, we presented a new method for selecting the optimal interface to access the Internet on mobile devices. We show that Available Bandwidth Estimation techniques available in the literature, e.g. Assolo, do not provide reliable results for modern mobile networks. To overcome this problem, we developed our own method to measure which of the two interfaces—WiFi or Cellular—performs better that executes quickly and uses small amounts of data.

The proposed method was implemented as a free application for Android and tested within a public LTE network. Our application also improves energy efficiency of data transfer on mobile devices by switching off the Internet links when they are not needed, at the cost of energy used to perform the measurements. We release an open source implementation of pik at `https://github.com/iitis/pik`.

We proposed a link selection method based on the available bandwidth, because it directly affects the web page and file download times perceived by the user. We leave the evaluation of other metrics that could be used for choosing between Cellular and WiFi, e.g. minimizing link latency, or maximizing the strength of the radio signal for further study.

Acknowledgment. This work was funded by the Polish National Centre for Research and Development, under research grant nr LIDER/10/194/L-3/11/:

project "Optimization and load balancing in next generation wireless networks", http://projekty.iitis.pl/zosb.

References

1. Aguehian, S.: Smart WiFi Toggler. https://play.google.com/store/apps/details?id=com.sebouh00.smartwifitoggler&hl=en Accessed: 2015-01-29.
2. Cisco Visual Networking Index: Global mobile data traffic forecast update, 2014-2019, Cisco Systems Inc (2014)
3. Witteeveen, D., Ward, S.: Global Mobile Consumer Survey 2013. http://goo.gl/P6xJvy Accessed: 2015-01-29.
4. Abinader, F.M., Almeida, E.P., Choudhury, S., Sousa, V.A., Cavalcante, A.M., Chaves, F.S., Tuomaala, E., Vieira, R.D., Doppler, K.: Performance Evaluation of IEEE 802.11n WLAN in Dense Deployment Scenarios. In: Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th, IEEE (2014) 1–5
5. Huang, J., Qian, F., Guo, Y., Zhou, Y., Xu, Q., Mao, Z.M., Sen, S., Spatscheck, O.: An in-depth study of LTE: effect of network protocol and application behavior on performance. In: ACM SIGCOMM 2013 Proceedings, ACM (2013) 363–374
6. Huang, J., Qian, F., Gerber, A., Mao, Z.M., Sen, S., Spatscheck, O.: A close examination of performance and power characteristics of 4G LTE networks. In: ACM MobiSys 2012 Proceedings, ACM (2012) 225–238
7. Dimatteo, S., Hui, P., Han, B., Li, V.O.: Cellular traffic offloading through wifi networks. In: Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, IEEE (2011) 192–201
8. Sankaran, C.: Data offloading techniques in 3gpp rel-10 networks: A tutorial. Communications Magazine, IEEE **50**(6) (2012) 46–53
9. Bennis, M., Simsek, M., Czylwik, A., Saad, W., Valentin, S., Debbah, M.: When cellular meets wifi in wireless small cell networks. Communications Magazine, IEEE **51**(6) (2013) 44–50
10. Gao, L., Iosifidis, G., Huang, J., Tassiulas, L.: Economics of mobile data offloading. In: Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, IEEE (2013) 351–356
11. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrell, L.: pathchirp: Efficient available bandwidth estimation for network paths. In: Passive and active measurement workshop. Volume 4. (2003)
12. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ACM (2003) 39–44
13. Goldoni, E., Rossi, G., Torelli, A.: Assolo, a new method for available bandwidth estimation. In: Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on, IEEE (2009) 130–136
14. Goldoni, E., Schivi, M.: End-to-end available bandwidth estimation tools, an experimental comparison. In: Traffic Monitoring and Analysis. Springer (2010) 171–182
15. Croce, D., Mellia, M., Leonardi, E.: The quest for bandwidth estimation techniques for large-scale distributed systems. ACM SIGMETRICS Performance Evaluation Review **37**(3) (2010) 20–25
16. Gorawski, M., Grochla, K.: Review of mobility models for performance evaluation of wireless networks. In: Man-Machine Interactions 3. Springer (2014) 567–577
17. Ookla: Speedtest.net Mobile. http://www.speedtest.net/mobile/ accessed: 2015-02-16.