Boz, Eren; Manner, Jukka

A hybrid approach to QoS measurements in cellular networks

Published in:
Computer Networks

DOI:
10.1016/j.comnet.2020.107158

Published: 08/05/2020

Document Version
Peer reviewed version

A hybrid approach to QoS measurements in cellular networks

Eren Boz[a,*], Jukka Manner[a]

[a]*Department of Communications and Networking, Aalto University, Maarintie 8, Espoo, Finland*

Abstract

Due to constantly increasing demand for mobile data, cellular network infrastructures running on limited radio spectrum are struggling to keep up. Constant monitoring and measurements are necessary to ensure service quality as saturated networks are not able to deliver consistent experience. However, measuring mobile networks at scale bears some fundamental problems. Although there are significant improvements in capabilities of mobile networks (e.g. bit rate, latency), measuring them is still rather complicated task compared to fixed networks given that in mobile networks, performance is a result of complex interaction between momentary cell load, adjacent cell interference, shadowing, fading, mobility and user device capabilities. The adoption of commercial 5G networks is expected to increase the variability even more as it depends on smaller cells. Active measurements that inject large amounts of traffic into the network for the sole purpose of measuring are costly in terms of both bandwidth and energy. Passive mechanisms are lightweight but miss the information of why a certain bit rate is received or sent by the end device. They can not tell whether the performance bottleneck is in the network or in the service itself. By combining active and passive measurements in a novel way, this study focuses on a hybrid measurement approach; that is cost-efficient, scalable and comparably accurate. In this paper, we develop a hybrid methodology where we passively measure incoming and outgoing bit rates and augment them with concurrent probe-based latency measurements to enable accurate network capacity estimations. We provide a model and heuristics to overcome issues related to radio access complications, capacity estimation, and optimization. Finally, we implement a prototype, deploy and evaluate it thoroughly to provide a proof-of-concept. We find that the proposed approach is not only highly accurate and a much efficient alternative to active measurements, but also superior in measuring user experienced quality.

Keywords: network measurement, cellular network, probe-based measurement, hybrid measurement

1. Introduction

Nowadays, it is quite uncommon to come across a person who does not own a connected device. Personal computers, tablets, smartphones are everywhere. Even watches are going smart along with other types of wearables. Every single one of them is connected, one way or the other, changing how we live our daily lives, solve basic problems and communicate. According to Ericsson [1], as of 2018, the number of smartphone mobile subscriptions has reached 5.1 billion. Projections show that this number will grow to 7.2 billion as of 2024. Not only the number of users increase but also demand per user is also growing considerably. The mobile data traffic has grown by 82% in 2018. Mobile data traffic is expected to grow 3.5x between 2018 and 2021. The main driver for traffic growth seems to come from video streaming. Video streaming applications expected to account for 74% of the total mobile traffic by 2024.

To study the performance of cellular networks, there exists a wide range of commercial tools, crowdsourcing based services and mobile apps [2]. The common aspect of these solutions is that they all provide some sort of on-demand active measurements with more or less fixed methodologies. Prescribed fixed methodologies tend to generate the same pattern of traffic and measure only a certain subset of network behavior. Cellular networks, by nature, are heterogeneous and highly dynamic. They exhibit complex behaviors depending on network load, mobility, radio environment, and traffic patterns. Let us imagine a smartphone traveling at a speed of 60km/h inside a bus, halfway done downloading a file, it's quite a dare to say it would download the other half of the file in a similar amount of time. It is even hard to say if the file download will ever be completed at some point in time. Simply, the bus might drive the phone away from the coverage.

The point is that in a highly variable and complex environment, a few numbers of measurement data points will not tell us much about the network. In order to achieve a statistical significance upon the measurements with respect to free variables affecting the resultant network performance, a great number of measurements should be drawn from varying scenarios and locations along the route of that particular bus. The question is whether these tools can collect that much measurement data to explain

*Corresponding author

Email addresses: eren.boz@aalto.fi (Eren Boz), jukka.manner@aalto.fi (Jukka Manner)

the expected behavior of a commercial cellular network in a crowded city. Let alone the number but, it is unknown if typical situations where measurements are done by using an active tool can be used to predict the actual usage scenarios at large. Consequently, it is difficult to claim that the measurements derived from these active network measurement tools depict the whole picture. On the other hand, the traffic injected into the network for bandwidth measurements introduces a considerably large amount of overheads both in terms of bandwidth and energy. Typically a set-up to reliably measure a high-speed connection with a TCP throughput active measurement requires data to be transferred at full speed for few seconds allowing enough time for TCP slow start and ramp-up to finish. This translates to 50-100 MB overhead for a modest 50+ Mbps Downlink, 20+ Mbps Uplink 4G connection, where many basic mobile data subscriptions offer a few GB per month. There are also various measurement concepts based on bursty packets trains [3], yet, all of these also generate a considerable amount of data, and only measure for a short while. So in a mobile environment where time, bandwidth and energy are scarce, the incentive for end-users to measure vanishes completely due to its costs.

To summarize, current mainstream active measurement methodologies are

- costly on bandwidth and energy;
- biased due to fixed traffic patterns and user measuring behavior; and
- not scalable enough to achieve a significant amount of measurements.

Passive methods are more lightweight on the network. By recording traffic bit rates and contextual information, such as radio parameters and locations, we can get more data on network performance. The challenge is that for the bit rates, we do not know the reason for them. For example, if a device is seen to receive 5 Mbps of traffic, is it the maximum speed of the server or the bit rate limited by the access network? This piece of information is critical to find any bottlenecks in the network performance.

Unlike pure active or passive measurements, we take a hybrid approach to best utilize the available capabilities of mobile platforms such as Android. Android allows us to track and sample mobile traffic rates along with contextual information e.g. time, location, operator, signal levels. However, such observations are weak by themselves to infer QoS metrics like network capacity or latency. Consequently, in this work, we develop a hybrid methodology where we utilize momentary traffic rates sampled from the cellular network interface that are augmented by concurrent probe-based latency measurements to enable accurate network capacity estimations. We provide a model and a set of algorithms and heuristics to tackle various issues (e.g. optimization, capacity estimation, radio access latency characteristics).

We show that this new approach is capable of continuously measuring without user initiation and achieving an accuracy as high as 95% with a cost of few kilobytes per measurement. Moreover, the methodology measures the actual experienced QoS by the user as it utilizes the mobile data usage itself.

Our contribution in this paper can be regarded as two-fold. First, we investigate probe-based measurement methodologies where we focus on challenges arising due to radio access complexities and then come up with a novel hybrid solution for QoS measurements in cellular networks that offers a scalable and cost-efficient way of estimating network capacity. Secondly, we implement a proof-of-concept for the proposed approach and empirically evaluate it via a large-scale study to reveal important aspects of the approach.

The remainder of the paper is structured as follows. First, we briefly discuss network measurement types to describe hybrid measurements in general. Then we cover a variety of probe-based measurement methods, their characteristics and why they are not directly usable in cellular networks. Following that, based on given facts, we describe a hybrid measurement model while discussing various aspects of it. We design, implement and present an Android application based prototype along with example measurements. Finally, we evaluate and validate our approach via an extensive empirical study while discussing the results in detail.

2. Background

The measurement methods which actively inject data to the network are classified as active measurements. Depending on the metric which is being measured, active measurements usually require cooperation from both ends, in end-to-end measurement scenarios. Metrics such as latency, jitter, throughput and packet loss can be measured easily using active measurement methods. In contrast to active measurements, passive measurements do not introduce data traffic to the network it is measuring. It is non-intrusive by nature. The methods of passive measurements involve capturing traffic generated by other users or applications. The process of capturing does not necessarily include all the bits that go through the point of measurement, but should be limited to only relevant information such as packet traces including headers of protocols that are of interest.

A measurement method does not have to be purely active or passive. Sometimes, it is more efficient to carry out measurements composing both active and passive elements to approximate a metric. Crovella and Krishnamurty [4] calls it fused measurements. An internet-draft by IETF names it as hybrid measurements [5]. According to the draft, a hybrid measurement is the combination of metrics derived from passive and active measurements to produce a measurement result. The example for a hybrid measurement could be measuring one-way delay to a set

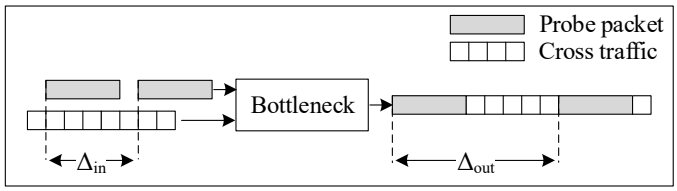


Figure 1: Probe Gap Model (PGM) illustration

of endpoints from passive measurements in which there is a frequent traffic exchange and it can be complemented with active measurements from endpoints with less frequent traffic to generate a delay map. That is to say, active measurements can be used to remedy the disadvantages of passive measurements.

2.1. Probe based measurement methodologies

Available bandwidth and capacity estimation on a path has beneficial use cases such as selecting better routes on an overlay network, QoS verification, and traffic engineering. In literature [3], various probe-based methodologies are proposed to measure capacity or available bandwidth in different fixed network settings such as single-hop or multi-hop. Essentially these methods can be classified into two main groups as probe rate model and probe gap model [6], based on the approach of measurement.

2.1.1. Probe gap model

The probe gap model (PGM) basically makes use of the time difference between the arrival of two successive probes at the receiver. The model denotes the time difference of consecutive probes sent in the sender side as Δ_{in} and the time difference of reception of those at the receiver side as Δ_{out}. Along with the assumption of single bottleneck (tight link) over the path of measurement, the relation of Δ_{in} and Δ_{out} constitutes a measure of congestion in the tight link as depicted in Figure 1, $\Delta_{out} - \Delta_{in}$ is the time it takes to transmit the cross traffic at the right link.

Furthermore, if the capacity information of the tight link is available, one can estimate the available bandwidth A using Equation 1.

$$A = C \times (1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}) \quad (1)$$

In a situation with a constant bit rate (CBR) cross-traffic, a small number of probe pairs would yield a plausible estimate by averaging samples calculated by the equation. However, cross-traffic rarely occurs at a constant bit rate. Moreover, most of the cross-traffic on the internet is bursty, so the real challenge for the PGM is to adapt the sending process of probe packets. Hence, PGM tools adaptively send a number of packet pair trains, which has several probes spaced variably to generate a rather unbiased estimation. Spruce [6], Initial-Gap-Increasing (IGI) [7] and Delphi [8] are example tools for estimating available bandwidth, which tries to overcome these challenges.

PGM tools typically have the advantage of low bandwidth usage to produce a measurement, but Lao et al. [9] shows that PGM can underestimate the available bandwidth of multi-hop paths even if there is a single tight link. Additionally, even though a single tight link assumption is plausible for most of the internet paths, the requirement for the knowledge of the capacity of the tight link is impractical. In conclusion, PGM tools fail to provide the desired reliability and practicality for most applications.

2.1.2. Probe rate model

Unlike the probe gap model, the probe rate model relies on the self-induced congestion on the network path to measure available bandwidth. Congestion on the tight link is going to cause the following packets on the path to experience additional queuing delays. Hence, injecting more traffic at a higher rate than available bandwidth as depicted by Figure 2, results in an increasing trend in one way delays (OWDs). PRM tools use this basic premise to measure available bandwidth.

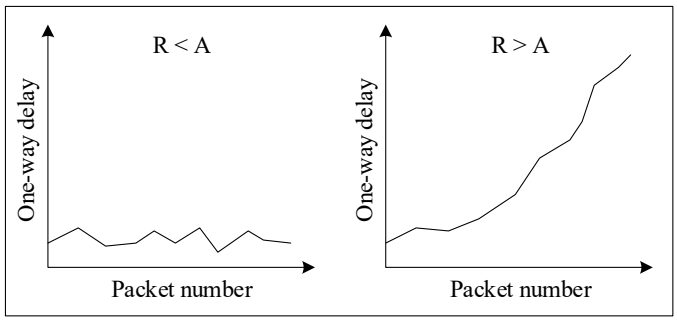


Figure 2: Probe Rate Model (PRM): the relation between one way delays and available bandwidth

Self-loading periodic streams (SlopS) methodology is a typical instance of the probe rate model [10]. In SlopS methodology, the source sends a fixed number of equal-size packets to the destination at a certain rate R. The receiver side monitors and reports the OWD trend back to the sender, so that the sender can statistically evaluate the trend and adjust the sending rate R accordingly. The adjustment is similar to a binary search if OWD is increasing, hence $R > A$, the rate R is decreased for the next iteration, if OWD is non-increasing in the case of $R < A$, R is increased. Once two successive sending rates are close to each other to a degree of ϵ, the algorithm estimates $A = R$. This type of approach requires a considerable number of iterations, especially in the case of highly varying available bandwidth.

Melander et al. [11] proposed a slightly different methodology than SlopS. The proposed methodology is built around the idea of trains of packet pairs (TOPP). In TOPP, the sender sends many packet pairs at gradually increasing rates, in other words gradually decreasing inter-packet interval Δ_S. The offered traffic rate at a packet pair, R_o is equal to L/Δ_S where L is the individual packet

size and Δ_S is the time it takes to receive a single probe packet. If $R_o > A$, the receive rate R_m is expected to be equal to R_o. Consequently, TOPP estimates available bandwidth as the maximum R_o where $R_o \approx R_m$.

Additionally, TOPP is able to estimate capacity of the narrow link in the path using Equation 2 where R_c is the rate of cross traffic. The capacity C can be estimated from the slope of R_o/R_m vs R_o.

$$R_m = \frac{R_o}{R_o + R_c} \times C \tag{2}$$

Even though TOPP has the benefit of multi-rate probing, it loses the temporal behavior of queuing caused by bursty traffic due to rather large spacing between the probe pairs.

2.2. Challenges for probe-based measurement methods in cellular networks

For practical purposes, it is beneficial to concentrate and summarize the reasons why probe-based measurements are challenging in the cellular network environment given the discussions up until this point. The common feature of all probe-based measurements is that they use the delays experienced by the probe to estimate the available bandwidth in the communication path. The assumption for this is that the change in the delay is only caused by the change in the available bandwidth. This assumption is awfully broken in cellular networks because the delay experienced by probe packets can be chiefly affected by interruptions caused by all sorts of handovers be it vertical or horizontal as well as Radio Resource Control (RRC) state changes. Additionally, varying radio conditions and signal quality can change packet loss characteristics of the channel and finally probe packets might be retransmitted which in return adds a multiple of retransmission base delay. Finally, in a loaded network along with the varying radio conditions, scheduling at the base station can immensely affect the delays experienced by each packet.

Secondarily, traffic patterns might affect how cellular networks behave. Moreover, the behavior dynamically might change in different operator's network, which depends on deployment and systems governing the radio resources. A tangible example is that RRC state transitions are typically triggered by what is called Buffer Occupancy (BO). BO, the increase in the buffered traffic at the base station, is an indicator of more traffic coming than the current service rate, hence it justifies the switch to a better RRC state and higher capacity channel. BO is also a configurable RRC parameter which is usually a value close to 1 KB [12].

Lastly, even if the cell has no load and the entire cell capacity is free for the user, it is very typical that, within the same RRC State, all radio resources for the maximum capacity are not allocated right away. Instead, parallel to BO idea, it might be gradually increased. The reason is that there might be a cost associated with deallocation and reallocation of the resources to possible incoming users. Spreading codes in WCDMA is an example of such a case.

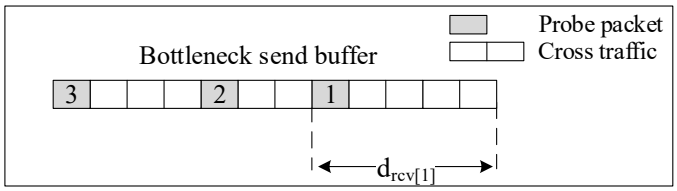


Figure 3: Base measurable model

In conclusion, for all these reasons, the delay and the capacity of the channel to a particular user systematically vary. To handle these variations, the probe methodology should be aware of the possible state of the network and ensure that samples derived from measurements belong to what is intended to be measured.

3. Hybrid model

In order to devise a cost-efficient, reliable and accurate method which is capable of measuring user link capacity in cellular networks with smartphones, we model a new approach to capture the best of both worlds, i.e. to be as cheap as PGM but as accurate as PRM at the same time.

3.1. Assumptions and requirements

The main requirement for the model is that the only single tight link is the user's access link which is the wireless link from base station to user device for cellular network case. Thus, the model measures the capacity of the tight link. The packet delay in the path from probe sender to the tight link is assumed to be near-constant and does not vary within a measurement. Also, the tight link is assumed to behave like FIFO per user traffic. The model requires the ability to measure the amount of cross-traffic between any two moments. All sorts of information related to cross-traffic such as packet count and packet sizes are useful, but not necessarily required.

3.2. Base measurable state

Figure 3 depicts the send buffer of tight link where a measurement can be done upon reception of the whole content by the user device. $d_{rcv}[n]$ denotes queuing delay experienced by the probe number n. Let $r[n]$ be the total number of bits received by the user at $t[n]$ which is the time of reception of the probe number n. Momentary link capacity can be simply estimated by Equation 3 where $d_{rcv}[n] > d_{rcv}[n-1]$, $(r[n] - r[n-1]) > 0$ and $(t[n] - t[n-1]) < d_{rcv}[n]$. That is the condition for probe number n and $n-1$ to be in the same queue and had some amount of cross traffic between them. Final estimation can be calculated by merging all viable samples within a meaningful time frame.

$$C(t[n]) = \frac{r[n] - r[n-1]}{d_{rcv}[n] - d_{rcv}[n-1]} \tag{3}$$

However, in practice, the exact queuing delays are not known to the receiver side. $d_{rcv}[n]$ can be approximated by $d[n] - min(d)$ where $d[n]$ is one way delay of probe number n and $min(d)$ is the minimum OWD which happens at the point where the packet sees the queue empty. Yet again, measurement of OWDs requires clock synchronization between sender and receiver which is impractical. Fortunately, absolute OWDs are not necessary, relative delays can be used to approximate queuing delays of probe packets, simply because $d_{rel}[n] = d[n] + clock_{offset}$. Naturally, this is the case only if the clock skew is negligibly small.

The base measurable state only describes the minimal conditions to obtain a measurement sample. The validity of the samples is greatly impacted by the implementation environment which includes all network aspects, user device hardware/software, probe sender/receiver implementation etc. So in short, it is essential to filter measurement samples at appropriate situations in order to improve the reliability of measurement results.

A simple framework for filtering measurement samples can be based on conditions enforced on measured values per sample. For example, the samples where there is no certain number of cross-traffic packets between consecutive probes can be filtered out, because it might simply be an anomaly caused by radio retransmissions. Similarly, if the probe packets received too closely in time relative to their separation at the probe sender, this situation can also be an anomaly caused by another element in the implementation stack. However, as we are aware that in packet networks the traffic is highly non-fluid. Hence, excessive filtering would result in inaccurate measurements.

One essential filter that could be utilized, is to establish a delay margin for minimal relative delays in the measurement sample. This delay margin can account for the delay variability for the reasons other than queuing in the tight link. At the same time the margin should be small enough so that it is possible to capture measurement samples at all, especially in cases where buffer allocated to the user at the base station is small.

3.3. Opportunistic approach and its challenges

The fundamental problem with passive measurements is that the measurement tool must be alert and running to observe the measurable state. In order to do so, the tool must continuously track the cross-traffic on a level that can detect the measurable state. The continuous tracking has an associated processing cost. Furthermore, on portable devices like smartphones, the battery is a way too valuable resource to waste.

On the other hand, the measurable state, in this case, is the state where the send buffer is grown enough that the communication channel is sending the traffic at its momentary achievable rates i.e. capacity to the user. In order to confirm such a state, the tool must be able to decide when to initiate a probe train and then inspect the probe delays to check if the measurable state as in Figure 3 is available. As it is well-known that most of the internet traffic is carried via TCP connections and most of these connections are short-lived; anticipating those short moments and appropriately injecting a probe pattern to achieve measurable state is an optimization problem of its own.

3.3.1. Inherent selection bias

Given that, as part of TCP's congestion control approach, the majority of internet traffic rates exhibit a gradually increasing pattern, especially shorter data transfers have a higher probability of observing lower speeds. Apart from the traffic pattern, our methodology can not measure what is not requested. So the distribution of the demand for speed, i.e. bulk data transfer will directly influence the measurements. Under normal circumstances, such influence is also likely to favor lower speeds due to bottlenecks in other points e.g. slow server. Additionally, burstiness of traffic is yet another point that possibly can shape the selection bias. Relatively short bursts combined with small buffers can cause probes to miss measurable states altogether. Yet again, in case of highly varying bandwidth where TCP fails to fill the pipe, queuing growths are more associated with decreasing speeds than increasing. Consequently, such factors result in a strong selection bias for our methodology.

3.3.2. Dummy-way vs Smart-way

Apart from the inherent selection bias, the logic of opportunity detection itself has the potential to introduce another kind of bias, because probes will be injected only when conditions dictated by the logic are met. In most dummy scenarios, for saving on probe costs, static traffic rate thresholds could be utilized to decide when to inject probe packets. However, if more parameters are available to sample such as packet size and counts, the logic can be improved by enforcing observed average packet sizes to be close to MTU. Thus, cross-traffic is more likely to be a bulk transfer that has more potential to saturate the link. Yet another improvement would be to maintain dynamic thresholds calculated from previous speed observations, given that link capacity has known statistical properties such as range of variation. Consequently, the design of opportunity detection is a trade-off point between cost and accuracy of the measurements.

4. Implementation

In this section, the prototype implementations of the hybrid model for realizing the goals of this study are described in details.

4.1. Probe server

Essentially, the measurement can be achieved using a simple ping-pong server including send and receive times-

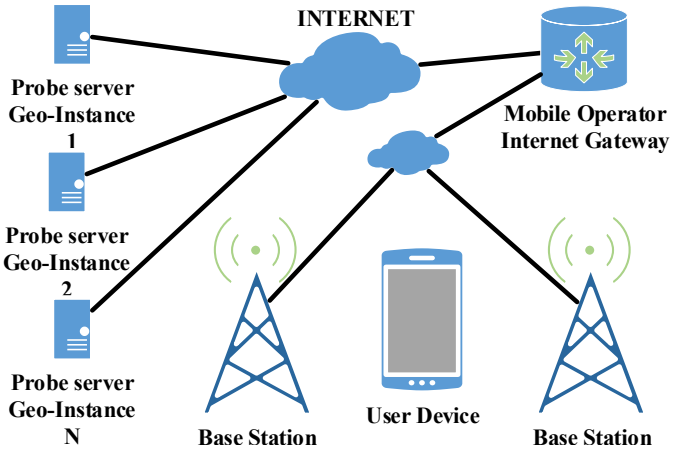


Figure 4: Probe servers network diagram

tamps in the packets. However, such an approach is challenged in our mission to achieve precision in probe intervals, mainly due to asymmetry in downlink and uplink. The additional variation in delays due to uplink decreases the timeliness of downlink probes. Furthermore, probe request per each probe is a waste of resources, especially in downlink measurements. Hence, the purpose of the probe server is to serve variable probe trains upon request so as to achieve measurable states at the user end. To ensure accurate queuing delay estimations, a set of geographically distributed servers should be utilized as shown in Figure 4. The user prototype application sends its probing request to the closest server by resolving geo-dns enabled domain name. Closer probe servers are desired to meet the near-constant delay requirement as well as to be able to quickly serve probe trains upon request.

The probe server has two different modes of functioning. For uplink measurements, the user application sends a UDP request packet per probe packet in various intervals where the server replies with a probe packet which includes a timestamp of the server's monotonic clock. Differences in the timestamps are used to calculate the queuing delays experienced by the requests in the uplink direction to ultimately collect measurable samples. On downlink measurements, the aforementioned optimized approach is followed because downlink traffic usages are much bigger and more frequent than uplink ones. Consequently, downlink measurable states occur much more often. For downlink probing, a subscriber model is followed. User application request a probing stream with a certain interval i, and the server sends a probe pair every i ms. User application needs to send keepalives to keep probes coming otherwise the subscription times out. Once the user application decides the probe stream is no longer necessary it can request it to be ceased.

A probe pair is two probe packets sent 20 ms apart. The value 20 ms is arbitrarily chosen by experimenting on Finnish commercial networks providing 3G and 4G connectivity, but each parameter in the implementation could be optimized per radio access technology to make the most out of it. The significance of probe pair separation is that the queuing delays need to grow above this value so that there is a chance of observing a measurable state. Assuming that there is enough cross-traffic to utilize the full capacity of the measured user, the chance of observing the transient measurable state is directly proportional to how often the probe pairs are sent. So this is a trade-off point between probe bandwidth overhead and the chance of obtaining measurement samples.

4.2. Client application

Android OS is an inherently eligible mobile platform for the implementation of our prototype measurement tool. It provides all the requirements demanded by our measurement model already on its developer framework. Moreover, it's the most popular platform which powers a great variety of devices.

The prototype is developed as an Android library which can be easily plugged into any other android application. The measurement service is implemented using Android background service [13] capability which enables application code to run in the background silently without intruding user activity. The background service is configured to start automatically on device boot up and run indefinitely.

The application does not continuously run the code, but sleeps in the background and listens for some trigger to wake up and start processing. The type of trigger that the application makes use of is called data activity of Android's phone state listener API [14]. The data activity trigger is fired whenever the user sends or receives any data from the cellular network which is the exact information that the measurement logic needs. As the code picks up the data activity trigger, it starts sampling the rate of incoming and outgoing traffic using traffic stats API [15]. However, in the uplink, the cross-traffic can not be sampled correctly as the traffic stats are derived from the amount of traffic passed the NIC/kernel boundary. Hence, the cross-traffic data is approximate for the uplink case.

The sampling is the phase where the application needs to decide if the probe from the server should be requested or not. We have implemented a dummy approach here by checking if downlink or uplink traffic is above some threshold. These thresholds are 100 Kbps and 200 Kbps respectively. Once the threshold for the downlink rate is reached, the application requests a probe stream with a 100 ms interval and periodically sends keepalives until the traffic rate goes below half of the threshold for 2 seconds. For the uplink, as covered in probe server discussion, the application sends probe echo request from the server every 60 ms. Yet again similarly if uplink traffic rate goes below half of the original threshold for 2 seconds the echo requests are stopped. These stop points are called idle phase. We define a ***session*** to be started by the sampling phase and ended by idle phase where a form of probe conversation occurs. For the time between the application records all the traffic rates as well as probe packets information

which includes sequence numbers, timestamps from both server and client application itself. Additionally, signal strengths, network technology, cell ID and location information that corresponds to the session are collected and bundled into a data structure. A list of these data structures piled until they reach a certain size, then the whole data is serialized and sent to the research server along with some detailed device information.

The resultant measurement service is very cost-efficient, in our prototype implementation probe sessions incur less than 10 Kbps overhead and less than 1% battery usage. The total bandwidth and battery usage by the service is naturally proportional to mobile data usage itself.

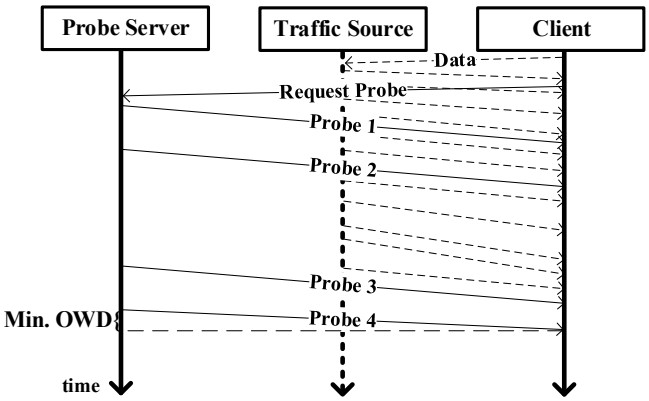


Figure 5: Sequence diagram for a short downlink measurement

The queuing delays of packets are estimated by calculating the relative delay of each probe packet. In the context of a session, the relative delay is calculated by using the probe packet with minimum offset as the difference between server send timestamp and application receive timestamp. This indicates that the particular probe packet was the fastest in that session. Typically, it is one of the last packets in the tail of the session where cross-traffic is almost non-existent as the session will become idle soon as shown in Figure 5. One interesting fact about the fastest probe packet is that it might not be the absolute fastest which is physically possible. This effect only changes the number of observed measurable states because the minimum offset is slightly bigger than the real value, so the calculated relative delays are slightly lower.

4.3. Bandwidth estimations

The samples which satisfy the base measurement state are determined by using their relative delays, arrival times. The total number of bytes received and the duration between measurement samples are calculated. These two components are essential results of the measurement sample which at least a bandwidth estimation can be produced by. Reduction of the measurement samples into speed dimension is not desired because the duration of a sample is crucial information regarding the calculation of a proper estimate based on weights by duration. The samples are known to carry the effects of packetization both in radio and IP layers as discussed.

However, these samples are not used right away but filtered by a set of conditions to eliminate radio access and implementation-specific effects. There was no delay margin filter imposed for this study, but to meet the model a filter is used to check the constraint of having at least a certain amount of cross-traffic between the arrival of probe pairs. This is especially desired in cases where there are few measurement samples so that the effects of interruptions in wireless communications (e.g. handovers, RRC state changes etc.) are removed. The parameters for this filter is arbitrarily chosen as 3 packets and 1000 bytes. Additionally, in the downlink, probe pairs received within 1 ms were filtered. These essential filters were applied to all samples and constitute a ***base***.

Lastly, in order to explore effects of a simple filter based on probe pair spacings, we separately define Equation 4 as means of further eliminating potential "noise" caused by radio hiccups and user-space processing. The packets can be buffered in the kernel before delivery to application, so fluctuations in the buffer result in different additional delays experienced by probe packets. These variations ultimately affect the calculated speeds. Such a filter restricts the measurement samples to be smoother where their spacing at the sender and receiver is similar. Hence, it aims to eliminate any samples affected by distortions in buffering or processing. The base measurements which are further filtered by this are referred to as ***filtered*** in our analyses.

$$\left|\frac{(t_{rcv}[n] - t_{rcv}[n-1])}{t_{snd}[n] - t_{snd}[n-1]} - 1\right| < 0.5 \quad (4)$$

5. Evaluation

The prototype was evaluated with an Android application which also featured active goodput measurements. A single probe server instance was deployed in a commercial cloud service in EU. We have collected data for roughly 2 years which resulted in about 9 million LTE and HSPA+ sessions each, from 3796 users, 260 operators and 522 distinct device models. However, the majority of the data originated from Finland and Finnish mobile operators. Consequently, This set-up resulted in a typical one-way delay of 25-30 ms excluding radio access network for the majority of measurement data collected. Additionally, we have data available for older 3G radio access technologies, but we omit them in our results as they are much less used. In this section, initially we visualize and investigate some example measurements of the prototype to get an idea on its typical behaviors. Following that we study the accuracy of the methodology by comparing it to active measurements performed by the same application. Finally, we investigate the measurement data to understand its properties at large.

5.1. Example measurements

A hybrid measurement is comprised of many data points to be able to make accurate estimations. Therefore, it is helpful to first visualize them. A plot schema as in Figures 6 - 9 with two Y-axes was designed to describe prototype measurements. Stars represent estimation samples where red and purple stars belong to samples filtered out by Equation 4 for downlink and uplink probes respectively. Vertical lines shows the weighted average (proportionate to sample duration) of corresponding samples so that we can see if there is any substantial difference between filtered samples and regular samples. Additionally, queuing delay experienced by probes over time can be seen by the black dots. The below figures show some example measurements detected by the prototype application from an LG Nexus 4 device with a 4Mbps limited HSPA+ connectivity. The network is known to be capable of achieving much higher speeds at the location of these measurements. Hence, 4Mbps is the expected correct value of the achievable download speeds. There wasn't an artificial limit enforced for the upload speeds. In Figure 6, it is observable that the tool is capable of detecting achievable rates even with lower average speeds shown by green download speed indicator. This is a result of burstiness in the traffic and a probe pair coinciding with this burst wave in the user send buffer of the base station. As discussed in the challenges of cellular networks, it also shows a clear example of how a radio interruption such as handover or RRC state change can affect measurement as indicated by probe delay peak of 150 milliseconds at the early phase of the session. The proposed simple algorithm seems to caught and filtered out a seemingly unreliable sample around the interruption, however, it also eliminated many more fine measurement samples. In Figure 7, a rather bursty upload session and how corresponding measurement turns out is showcased. Note that relative delays refer to ongoing measurement direction. Thus, while upload traffic is below the measurement threshold, relative delays belong to downlink probes.

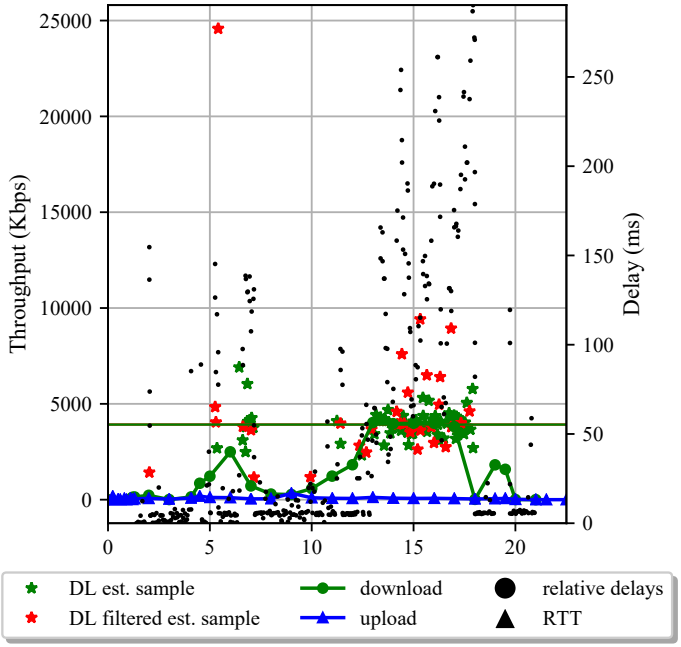


Figure 6: A typical web browsing session

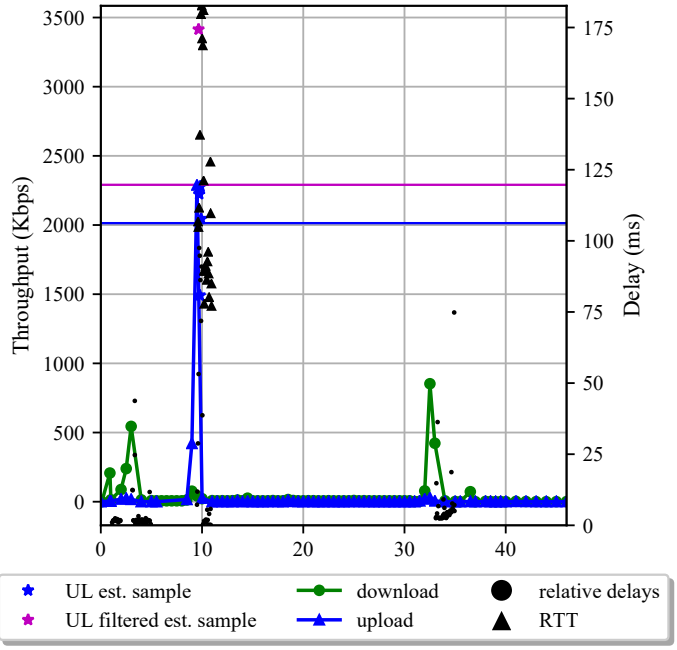


Figure 7: A brief upload measurement session

Figure 8 shows the measurement of YouTube video streaming session which has its particular bursty traffic pattern. Initially, the client buffers a certain amount of data to ensure the smooth playout of the media. Subsequently, we observe periodical bursts and an example of coincidental interaction of probes and bursty traffic patterns.

We observe that our algorithm managed to carry out highly accurate estimations in these controlled example measurements by tackling varying traffic patterns and cellular network specific issues. However, we also observe that having a low link capacity allowed us to get an ample amount of samples in these example measurements to make accurate predictions. We certainly need to further investigate the accuracy for a high variety of cases to confirm the overall accuracy of the approach.

5.2. Accuracy analysis

In order to investigate the accuracy of the hybrid measurements, first, we have analyzed how active goodput measurements compare when they act as a cross-traffic to generate hybrid measurement samples as in Figure 9. Then we visit the differences in statistical properties of hybrid measurements as opposed to active measurements.

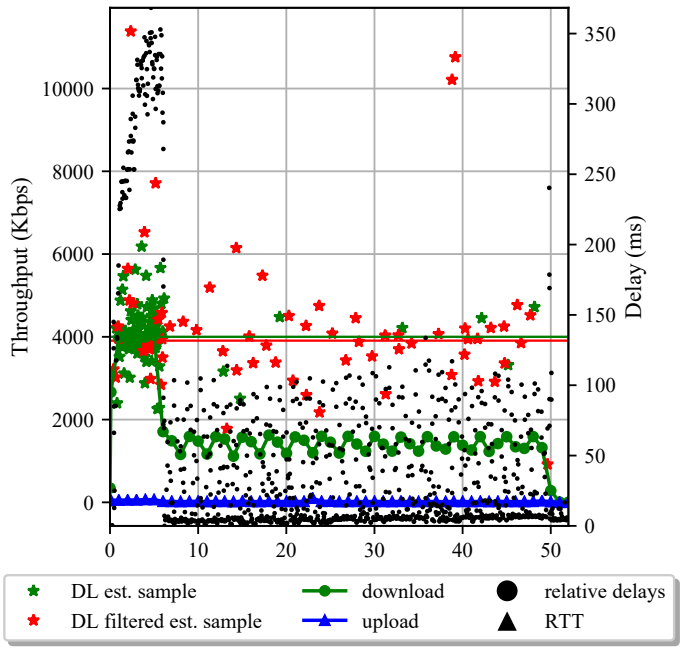


Figure 8: A youtube video viewing session

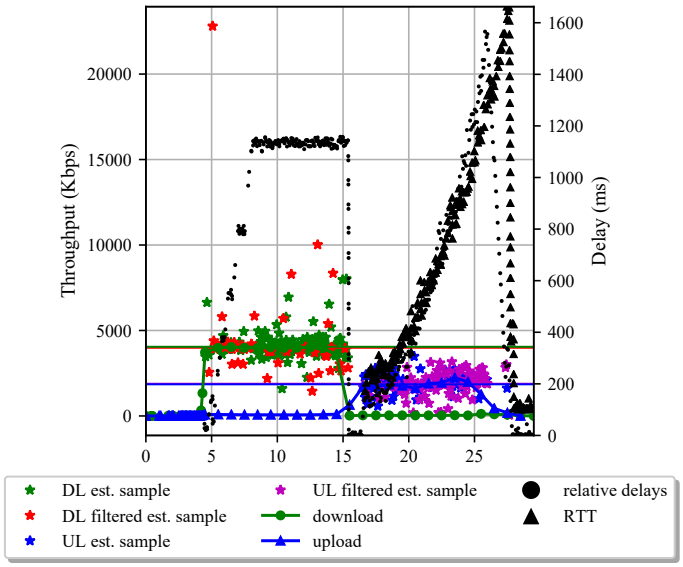


Figure 9: A hybrid measurement session corresponding to an active measurement as cross traffic source

5.2.1. Ground truth

Firstly, we would like to acknowledge that the most granular analysis of accuracy would be to compare base station data to momentary estimations. However we can not practically carry out such experiments on a large scale with multiple operators, hence our experiment is set up to explore it via a proxy. Hybrid measurements as implemented based on NIC traffic counters measure IP throughput, however, the experimental setup does not have such active measurement but expectedly closest proxy that we can utilize is active goodput measurements. Such comparison is certainly weak in the aspect that TCP can not guarantee full link utilization especially in the cases of unstable connectivity and poor buffer policy leading to starvation. However, goodput values provide a definite lower bound for both TCP and IP throughput. Hence the comparison allows us to roughly explore the typical ceiling for the accuracy figures in different scenarios. Also, goodput comparability is of great interest to understand how our methodology relates to typical measurement tools out there.

$$ErrorRatio = \frac{0.97 \times Estimation_{t>5s}}{Goodput_{t>5s}} - 1 \qquad (5)$$

We use a factor of 0.97 to account for approximate overhead of IPv4 and TCP to make the comparison more sensible. To avoid the slow-start portion of the active measurements, we only compare the second half of the 10-second measurements. Equation 5 defines an arbitrary signed error metric to be able to assess the occurrences of overestimation and underestimation. Such a definition also helps us to make sense of the direction of average errors where underestimations and overestimations might cancel each other out.

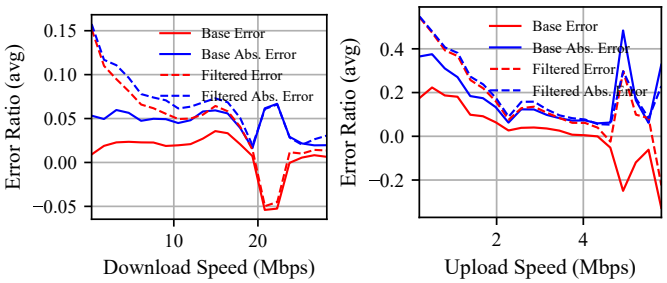
5.2.2. Estimation accuracies based on goodput comparison

At this point, we have two separate datasets to work with. ***Dataset A*** comprised of the hybrid measurements which we inspected its characteristics in the wild. ***Dataset B*** is composed of active goodput measurements and associated hybrid measurement sessions. To further refine our comparative analysis we limit the Dataset B to only include measurements where goodput has more than 4 seconds non-zero samples and estimation has more than 0.2 seconds. Such elimination is useful to limit our comparison to relatively stable connectivity which includes a considerable period of saturation. However, the difference in the sample sizes would still lead to inflated error ratios where the throughput variation is high. Additionally, we evaluate the simple filtering condition as defined in Equation 4 to explore the effects of filtering based on probe arrivals.

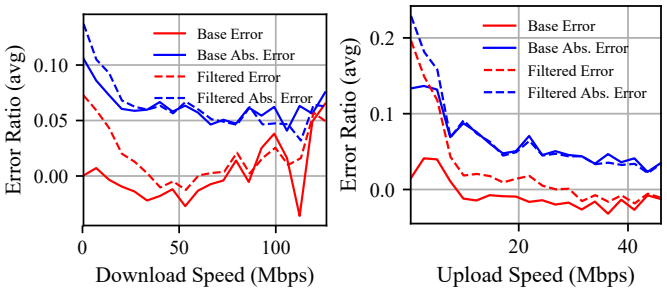
Figure 10 shows typical accuracy numbers changing with the speed. We observe that filtered measurements usually have higher error rates especially at lower speeds where they tend to overestimate. Not only they increase error rates but they also decrease the number of measurement samples which adds to sampling error in general. The uplink measurements experience higher error ratios as the expected result of NIC stat reading issues described in section 4.2. Another interesting observation is spikes in underestimation on physical channel limits such as 21.1Mbps for HSPA+. This can be attributed to sampling error gaining a direction in such cases as samples can not exceed the maximum transmission speed significantly.

5.3. The effects of selection bias

Finally, we take a look at the visible effects of selection bias in our hybrid measurements due to the opportunistic



(a) DL estimations (HSPA+) (b) UL estimations (HSPA+)



(c) DL estimations (LTE) (d) UL estimations (LTE)

Figure 10: Accuracy of estimations based on active measurement comparisons

approach. To be able to inspect and compare the results, we produced empirical distribution plots of our datasets in various cases. It's crucial to note that although datasets are derived from the same user base, they certainly do not belong to the same population in a statistical sense as we have not done any stratification or matching in any direction to make them more comparable. Dataset B, where active throughput measurement acts as cross-traffic, is characterized by a comparatively very small number of measurements featuring its own biases due to user measuring behavior. On the other hand, Dataset A is dominated by the selection bias of the opportunistic approach where we expect lower speeds to be caught more. Figure 11 confirms such expectation on a very high-level comparison. In spite of estimations in Dataset B are accurate on predicting goodput, we observe that in Dataset A measurements resulted in much lower numbers for the reasons discussed in section 3.3.1.

On a more particular level, Figure 12 and Figure 13 provides the same comparison for Finnish operators. Although the comparison is still subject to the same issues as before, in this case, we observe the effect of bias in determining the rank of operators as we see in the case of Operator 1 and 3. As the selection bias interacts with the speed demand characteristics of the user population, it might have varying effects on different operators. Ultimately such comparisons must be carried out on matched measurement datasets on features affecting the performance of the network such as location, device, time of the day etc. Nevertheless, our results provide an initial insight on large scale comparability of hybrid measurements and active goodput measurements.

5.4. Hybrid measurements in the wild

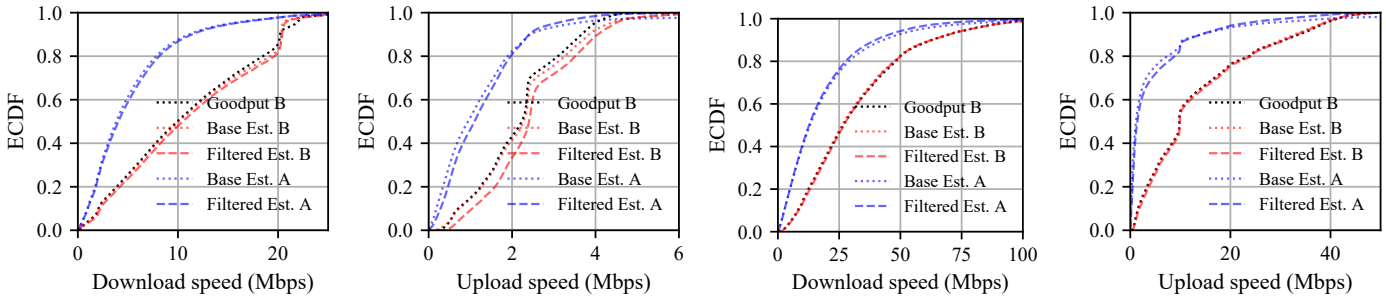
Extensive empirical data allows us to look at various aspects of mobile data usage in-depth, but firstly it is essential to provide some descriptive statistics to have a rudimentary overview of the overall data. The most basic property that we need to understand what a session looks like in terms of our parameters. In Figure 14, it can be seen as intuitively expected, the session lengths as defined in section 4.2, are exponentially distributed for the most part. The difference in HSPA+ and LTE shows that faster connectivity results in even shorter sessions which can be interpreted as a demand for speed, in general, is higher than what is provided by the operators.

Figure 15 provides two types of the overall distribution of bandwidth estimations in the wild. Note that the measurements in this first part are only filtered with an essential filter in order to provide a detailed analysis. In the normal case, a single average number is calculated for a single session regardless of the number of measurement samples. In the weighted case, the averages are weighted with the total duration of samples within the session. In the downlink, this creates a considerable difference between two because the weighted calculation is heavily biased towards lower speeds at least due to the fact that slower connection results in longer measurements. However, unintuitively, we observe the reverse in the uplink direction which is revealed to be due to how uplink capacity changes over time. For both directions, the figures seem to be well within access technology limitations. However, we observe an interesting plateau at 10 Mbps for LTE uploads. As in the case of our 4 Mbps downlink limited example measurements, this is caused by an operator enforced subscription limits in Finland. It's known that a certain operator in Finland provides up to 50 Mbps downlink and 10 Mbps uplink subscription. Yet the same can not be observed in the downlink which could be due to several reasons, but most likely because those subscribers could not experience upper limit as suggested by the median of LTE downlink measurements being much lower than 50 Mbps.

The other important aspect of our measurement approach is the buffering and queuing aspects in the narrow links that we are interested in. Figure 16 shows the average queuing delays estimated by sessions. Such information lets us know what ratio of the session actually experience a form of congestion and eligible for the generation of a bandwidth estimation measurement.

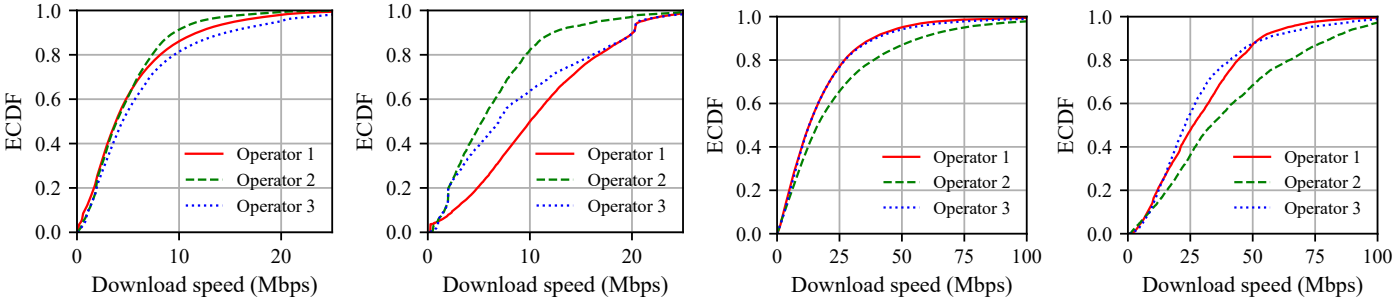
5.4.1. Characteristics of probe trains

As discussed for convenience and consistency, we have decided to use fixed intervals for probing. In the downlink, Figure 17 shows that precisely fixed spaced probes sent by the sender were received in quite varying patterns. Firstly, the frequent 10ms peaks for HSPA+ is as expected due



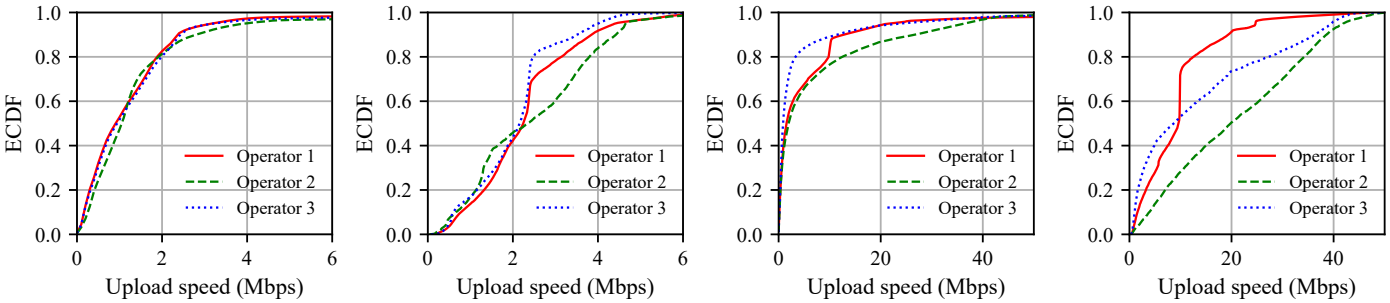
(a) DL estimations (HSPA+) (b) UL estimations (HSPA+) (c) DL estimations (LTE) (d) UL estimations (LTE)

Figure 11: Cumulative distribution comparisons of datasets and estimations



(a) Dataset A (HSPA+) (b) Dataset B (HSPA+) (c) Dataset A (LTE) (d) Dataset B (LTE)

Figure 12: Cumulative distributions of estimated download speeds of Finnish operators



(a) Dataset A (HSPA+) (b) Dataset B (HSPA+) (c) Dataset A (LTE) (d) Dataset B (LTE)

Figure 13: Cumulative distributions of estimated upload speeds of Finnish operators

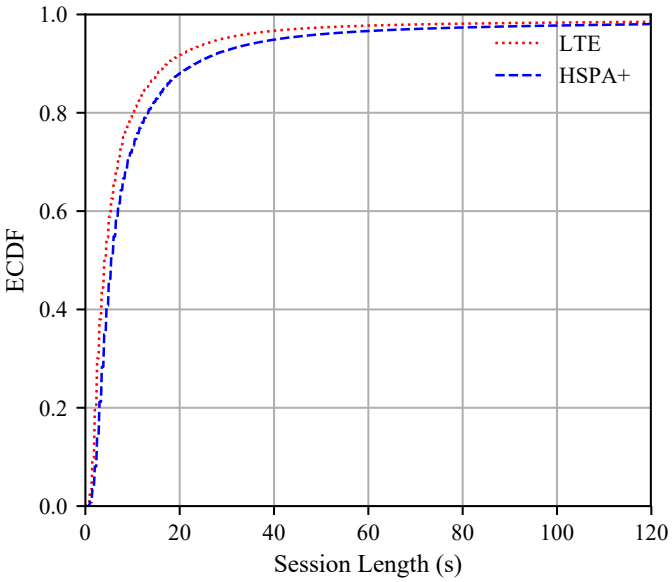
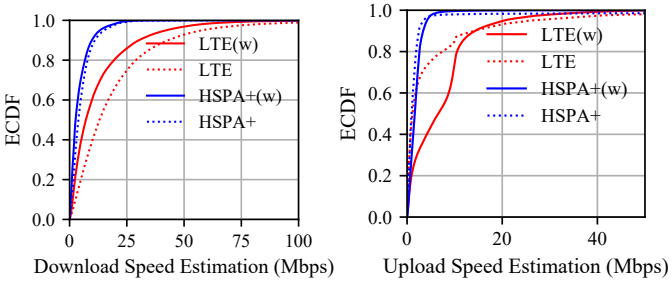
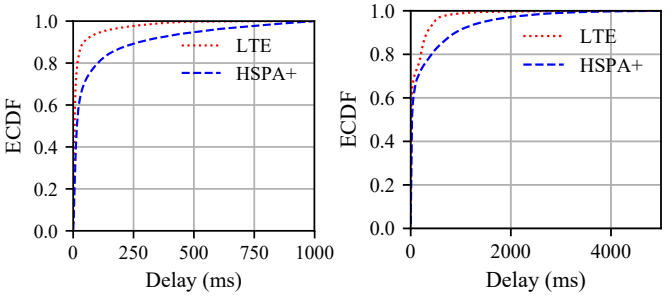


Figure 14: Cumulative distributions of session lengths



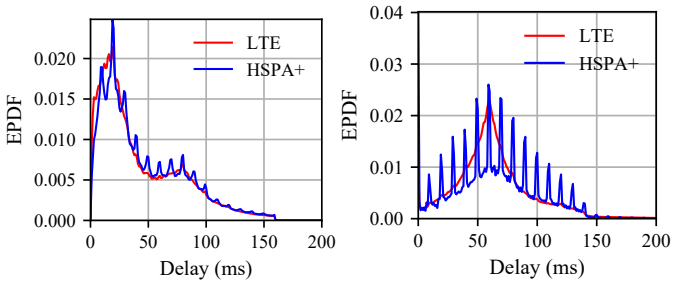
(a) Downlink (b) Uplink

Figure 15: Cumulative distributions of estimations



(a) Downlink probes (b) Uplink probes

Figure 16: Cumulative distributions of delay averages

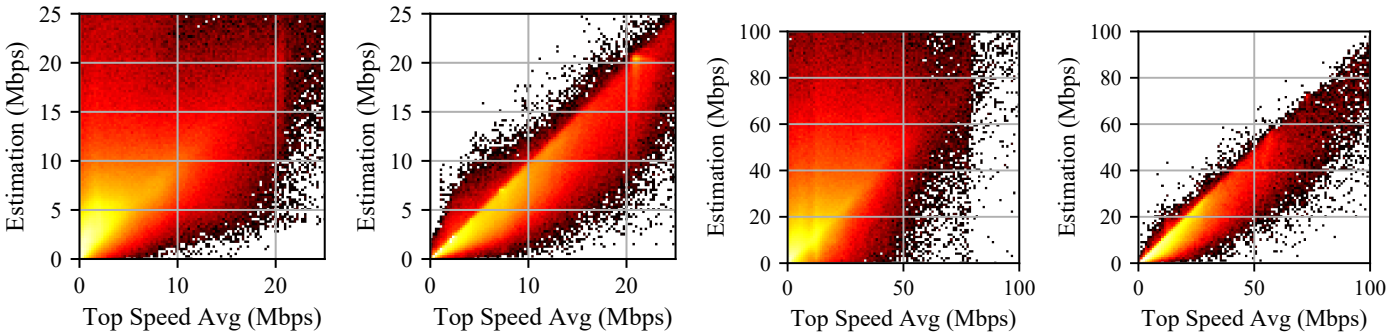


(a) Downlink probes (b) Uplink probes

Figure 17: Receive separation distributions of estimation probes

to 10 ms TTI, whereas LTE features a 1 ms TTI. The secondary peaks are naturally due to sending patterns. By design, the sending frequencies are roughly equal for both 20 ms and 80 ms intervals but since figure only includes samples generating an estimation, 80 ms spaced probes are naturally less likely to generate estimation due to the queuing process itself. Figure 17 also pictures the same behavior for the uplink case, only this time it includes a single uniform interval of 60 ms apart probe packets.

The differences in the sending and reception of the probes are mostly associated with the relative change in the link capacity and traffic arrival rates. Hence, filtering estimation samples solely based on this difference can be expected to result in more biased measurements towards stable states. Figure 19 specifically shows the characteristics of changes in the sending and arrival of probe pairs. The first clear observation is expanding probe intervals are associated with higher speed, whereas contracting ones are significantly lower. Secondly, both in downlink and uplink we observe prominent peaks. These are mainly caused by the implementation environment where we sample cross traffic at the NIC statistics level. So the accuracy of the sampling is heavily affected by the NIC and kernel interaction. Presumably, in the downlink case, NIC seems to deliver received packets as soon as few milliseconds, whereas in the uplink side we observe a different phenomenon. Especially in the case of HSPA+, the peak is extraordinarily large. In uplink cross-traffic sampling, the additional error is expected due to radio firmware buffer. Guo et al.[16] point out that radio firmware buffers exhibit adaptive behavior to accommodate higher speeds and prevent starvation in the process. Our figures show that there is a process of buffering with 20 ms intervals causing irregularity in estimation distributions. Additionally, in Figure 20b some outlying patterns are possibly caused by our single interval probe pattern coinciding with this cycle resulting in highly inaccurate measurements. So we can conclude that uniform probe patterns are prone to such errors in the uplink. It can be stated that simple filtering solely based on sending and arrival spacing is a possible source of error more than it can fix. For that reason, there should be a holistic approach to filtering rather than per sample treatment.



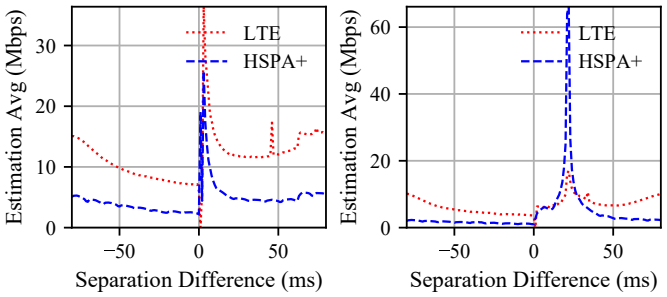
(a) Sample size up to 200 milliseconds (HSPA+)

(b) Sample size more than 1000 milliseconds (HSPA+)

(c) Sample size up to 200 milliseconds (LTE)

(d) Sample size more than 1000 milliseconds (LTE)

Figure 18: Log-normalized heatmaps for downlink estimations compared to achieved top speed averages(0.5s) of measurement sessions



(a) Downlink probes

(b) Uplink probes

Figure 19: Estimation averages of varying Receive-Send separation differences

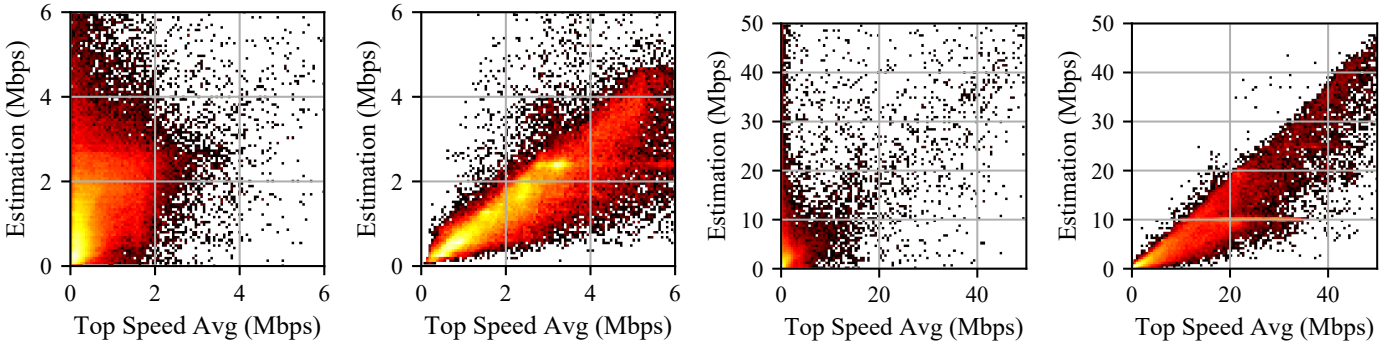
5.4.2. Characteristics of estimations

As discussed before the majority of the sessions are very short-lived. Also considering the time it takes to detect an opportunity and initiating a probe injection takes additional time where it's possible to miss measurements. Figure 21 reveals that the majority of downlink measurements yielding higher speeds take place in the first few seconds. We observe that longer sessions are associated with lower speeds where provided service lags behind the demand. In the uplink, a slightly different phenomenon occurs. It takes some time for uplink throughput to reach its peak values. Such behavior is expected due to radio resource behavior in cellular networks as they allocate uplink resources reactively to buffer status reported by the UE. Yet, the other aspect, in this case, is that the uplink traffic patterns that generate measurements are usually big bulk data transfers such as uploading multimedia files. Given that average upload speeds are much lower, the majority of the measurements are derived from relatively longer sessions. Also, note that the higher parts of the time axis in figures are characterized by a smaller number of measurement due to the distribution of session lengths.

The same problem could be further inspected from the perspective of the queuing delays. Figure 22 not only supports our conclusion on higher speed measurements but strongly points out to shortcomings of our probing parameters. In the downlink, extremely high speeds were mostly detected by probes with even smaller queuing delays than 20 ms. Even though results could have been amplified by implementation errors such as NIC buffering, it still points to a clear trend. Hence, such results call for a combination of more frequent probing and faster initiation to be able to detect bursty or short-lived measurement opportunities. Finally, Figure 18 and Figure 20 visually provide a means of investigating burstiness, sampling errors and variability characteristics of the measurements by looking at the relation of top achieved throughput average and estimation average in the session. Note that, we opted for log-normalization to bring about the extent of marginal behaviors. As can be seen from the figures, small sample sizes tend to exhibit variable behaviors which may be associated with burstiness in the traffic pattern especially in very short sessions. Since the sample size that comprises the measurement can even be as small as a few milliseconds, the reliability of these measurements is first bounded by sampling error and then the accuracy of the sample itself. The behavior of longer measurements is typically free from the effects of burstiness assuming that they were able to saturate the link for the most part. Such behavior results in almost no measurements in the upper diagonal. However, not so much in the case of HSPA+ uplink as its dominated by potential implementation errors of buffering and inaccurate cross-traffic estimations combined with high TTI.

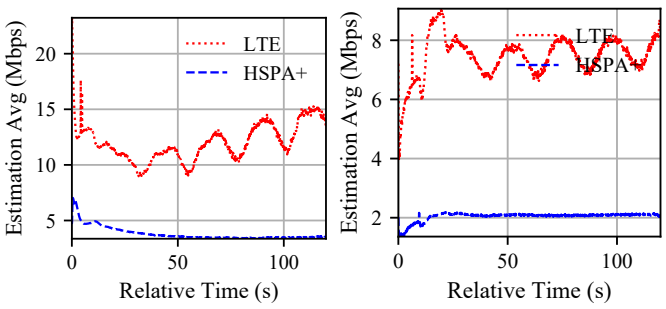
6. Discussion

Accuracy figures and selection biases have the potential to significantly deviate among different cellular network deployments. This is majorly due to different mechanisms such as Adaptive Queuing Management(AQM) or scheduling disciplines that might alter the traffic arrival characteristics and buffering process. As a result, in such cases, sanity checks for those issues should be carried out to make sure that they don't systematically skew the mea-



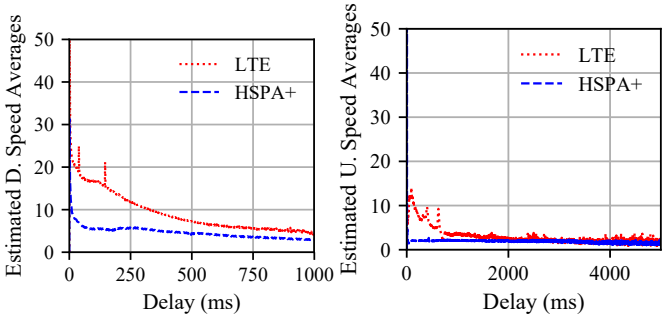
(a) Sample size up to 200 milliseconds (HSPA+)
(b) Sample size more than 1000 milliseconds (HSPA+)
(c) Sample size up to 200 milliseconds (LTE)
(d) Sample size more than 1000 milliseconds (LTE)

Figure 20: Log-normalized heatmaps for uplink estimations compared to achieved top speed averages(0.5s) of measurement sessions



(a) Downlink estimations
(b) Uplink estimations

Figure 21: Estimation averages of relative session time



(a) Downlink probes
(b) Uplink probes

Figure 22: Estimation averages of varying probe delays

surement results. Additionally, in this study majority of the measurements are characterized by a short distance to probe server which results in relatively accurate queuing delay estimations throughout the dataset. Consequently, it is naturally expected for measurement results to become less reliable when the assumptions of the model are not properly met.

Although, in this paper we mainly focused on hybrid bandwidth estimations, probing data can also be used to estimate QoS metrics such as RTT, loss, jitter with little effort. As opposed to commonplace tools, those metrics can be explored for different traffic patterns. For example, jitter originating from the cellular network can be analyzed for cases of constant or variable bit rate real-time communications such as VoIP. Furthermore, loss spikes combined with cross-traffic observations can be used to infer user experienced connectivity losses to approximate availability.

Another important remark is that our results belong to static probe patterns and a dummy way of detecting measurement opportunities which typically results in a few megabytes measurement overhead per day. The overhead can be hugely improved by smarter deployments of the methodology where probing is seldom performed during network usage. We have briefly experimented using a dynamic exponential backoff method for the probing threshold based on maximum achieved speeds in previous measurements. There was about 30 fold improvement in overhead. However, such fundamental changes would certainly result in different selection biases. Consequently, comparative studies of the measurements must be based on careful statistical modeling of the data. Furthermore, given that our methodology bears the potential to gather a large amount of data, a machine learning approach can be taken to explore the predictability of the bandwidth using additional data points such as location, device, operator, signal levels, cell ID, time of the day, etc.

7. Related work

In 2006, Diaz et al. visions and proposes the architecture of a crowdsourced measurement system that is made very possible today by the fact that almost most of the devices include a GPS chip. The architecture is simply composed of soft-agents installed to user terminals that run measurements at appropriate times. Measurement results along with the location, time and service type information are fed back to a server where the information is processed. Through analysis of the centralized information, the system would be able to provide real-time network diagnosis, service performance degradation monitoring and service availability monitoring [17]. Essentially, this type of architecture is used by many measurement services as it is the most logical and convenient one for the purpose.

Yao et al. conducted an eight-month-long study to inspect bandwidth predictability in a 3G cellular network. Experimenters drove through the same 26 km route many times while doing measurements at different points of time, to account for a consistent realistic scenario of mobility. Bandwidth measurements are carried out using a fast converging probe-based measurement to achieve cost efficiency and granularity. They have found out that using past measurement data along with the location information is a good predictor for the bandwidth [18].

Yet on another measurement study called Mobile Internet Services Test, Wittie et al. approached the issue from the perspective of mobile application developers, on the fact that underlying network characteristics directly impact the user experience of an application. Considering the fact, that the user experience and usability are the most critical aspect of a mobile application, it is important to understand the issues regarding the interaction of network and software. In order to characterize cellular data network performance as it is experienced by individual mobile devices, they developed a mobile application to measure interactions between network technology and a variety of mobile devices and platforms [19].

Gember et al. conducted a study to obtain in-context measurements on the premises that to quantify user experience, the relevant measurements need to be carried out during the relevant contexts. They found out that recent usage of network due to user activity, macro-environment (e.g. indoors and stationary), micro-environment (e.g. in the user's hand) matters as the context of measurements and changes the results [20]. Mobiperf is also another related measurement study by Huang et al.[21]. One of the significant contributions by Mobiperf is that they have introduced a RRC State Machine inference methodology in their applications to discover parameters of network state changes. Using the parameters they were able to assess performance, energy consumption characteristics of 3G and 4G networks [22, 23].

Related studies mentioned so far, try to address the problems stated to some extent, except the cost-efficient bandwidth measurements. Gping-pair [24] uses a PGM approach in cellular networks to achieve low-cost coarse bandwidth estimations. However, they try to capture complexities related to cellular networks only by an asymmetric downlink and uplink model. Consequently, this approach produces unreliable estimations. Gping-pair also uses ICMP based approach to eliminate the need for a probe server and uses the first reachable gateway via increasing TTL method for RTT measurements. That is also a potential source of error as ICMP is known to be highly rate-limited in the internet routers which can distort latency measurements [25].

8. Conclusion

The mobile-first era is here and demand for mobile data is growing stronger than ever. Regardless of the improvements in capabilities of mobile networks, measuring them is still rather complicated task given the nature of mobile communications. The resultant performance of a connection in mobile networks is a complex interaction between momentary cell load, adjacent cell interference, shadowing, fading, mobility and user device capabilities. Along with upcoming commercial adoption of 5G networks, heterogeneity and variability in radio access is expected to grow even further due to the fact that improvements heavily depend on elements such as smaller cells, fragile millimeter-wave bands, etc. Consequently, accurate quality mapping requires frequent measurements with both time and space diversity that's representative of how the networks are actually used by end-users.

In this paper, we proposed a hybrid QoS measurement model and methodology that has the potential to meet such challenges. Furthermore, we prototyped a proof-of-concept Android application. Finally, we carried out a large-scale evaluation study that has shown how not only continuous QoS measurements that provides cost-efficient network capacity measurements, but also mobile data usage characteristics can be gathered at scale with such little overheads. Our work can be readily and easily integrated into mobile platforms as a service and can be exposed to application developers to improve the networking and user experience aspects of their apps. Android network capabilities API that provides coarse bandwidth information is a strong example of such a use case [26].

Finally, looking into the future, having much cheaper measurements along with the possibility to dynamically control the sampling enables us to develop better approaches to crowdsourcing network measurements as well. It's entirely possible to develop a centralized approach that selectively optimizes measurements based on information gain principles. This way we can rather "crowdsense" network quality instead of unnecessarily measuring the same conditions (e.g. location, time, technology, operator, signal quality, device) repetitively.

Acknowledgment

This work was supported by the EMERGENT Project (http://emergent.comnet.aalto.fi/).

References

[1] Ericsson, Ericsson Mobility Report 2019, https://www.ericsson.com/49d1d9/assets/local/mobility-report/documents/2019/ericsson-mobility-report-june-2019.pdf, [Referenced: 2019-09-11] (June 2019).

[2] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, A. Vecchio, Smartphone-based crowdsourcing for network monitoring: opportunities, challenges, and a case study, IEEE Communications Magazine 52 (1) (2014) 106–113.

[3] R. Prasad, C. Dovrolis, M. Murray, K. Claffy, Bandwidth estimation: metrics, measurement techniques, and tools, Network, IEEE 17 (6) (2003) 27–35.

[4] M. Crovella, B. Krishnamurthy, Internet Measurement: Infrastructure, Traffic and Applications, John Wiley & Sons, Inc., New York, NY, USA, 2006.

[5] B. Trammell, L. Zheng, S. Silva, M. Bagnulo, Hybrid Measurement using IPPM Metrics, IETF (2014).
URL http://tools.ietf.org/html/draft-trammell-ippm-hybrid-ps-01

[6] J. Strauss, D. Katabi, F. Kaashoek, A measurement study of available bandwidth estimation tools, in: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ACM, 2003, pp. 39–44.

[7] N. Hu, P. Steenkiste, Evaluation and characterization of available bandwidth probing techniques, Selected Areas in Communications, IEEE Journal on 21 (6) (2003) 879–894.

[8] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, R. Baraniuk, Multifractal cross-traffic estimation, in: Proc. of ITC specialist seminar on IP traffic Measurement, 2000.

[9] L. Lao, C. Dovrolis, M. Sanadidi, The probe gap model can underestimate the available bandwidth of multihop paths, ACM SIGCOMM Computer Communication Review 36 (5) (2006) 29–34.

[10] M. Jain, C. Dovrolis, End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput, in: ACM SIGCOMM Computer Communication Review, Vol. 32, ACM, 2002, pp. 295–308.

[11] B. Melander, M. Bjorkman, P. Gunningberg, A new end-to-end probing and analysis method for estimating bandwidth bottlenecks, in: Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE, Vol. 1, IEEE, 2000, pp. 415–420.

[12] P. H. Perala, A. Barbuzzi, G. Boggia, K. Pentikousis, Theory and practice of rrc state transitions in umts networks, in: GLOBECOM Workshops, 2009 IEEE, IEEE, 2009, pp. 1–6.

[13] A. Developers, Android services, https://developer.android.com/guide/components/services, [Referenced: 2019-09-11].

[14] A. Developers, Android phone state listener, http://developer.android.com/reference/android/telephony/PhoneStateListener, [Referenced: 2019-09-11].

[15] A. Developers, Android traffic stats, https://developer.android.com/reference/android/net/TrafficStats, [Referenced: 2019-09-11].

[16] Y. Guo, F. Qian, Q. A. Chen, Z. M. Mao, S. Sen, Understanding on-device bufferbloat for cellular upload, in: Proceedings of the 2016 ACM on Internet Measurement Conference, IMC '16, ACM, New York, NY, USA, 2016, pp. 303–317. doi:10.1145/2987443.2987490.
URL http://doi.acm.org/10.1145/2987443.2987490

[17] A. Diaz, P. Merino, A. Gil, J. Munoz, x-appmonitor muagent: a tool for qos measurements in cellular networks, in: Wireless Communication Systems, 2006. ISWCS '06. 3rd International Symposium on, 2006, pp. 343–347. doi:10.1109/ISWCS.2006.4362316.

[18] J. Yao, S. S. Kanhere, M. Hassan, An empirical study of bandwidth predictability in mobile computing, in: Proceedings of the Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH '08, ACM, New York, NY, USA, 2008, pp. 11–18. doi:10.1145/1410077.1410081.
URL http://doi.acm.org/10.1145/1410077.1410081

[19] M. Wittie, B. Stone-Gross, K. Almeroth, E. Belding, Mist: Cellular data network measurement for mobile applications, in: Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on, 2007, pp. 743–751. doi:10.1109/BROADNETS.2007.4550508.

[20] A. Gember, A. Akella, J. Pang, A. Varshavsky, R. Caceres, Obtaining in-context measurements of cellular network performance, in: Proceedings of the 2012 ACM Conference on Internet Measurement Conference, IMC '12, ACM, New York, NY, USA, 2012, pp. 287–300. doi:10.1145/2398776.2398807.
URL http://doi.acm.org/10.1145/2398776.2398807

[21] J. Huang, C. Chen, Y. Pei, Z. Wang, Z. Qian, F. Qian, B. Tiwana, Q. Xu, Z. Mao, M. Zhang, et al., Mobiperf: Mobile network measurement system, Technical Report. University of Michigan and Microsoft Research (2011).

[22] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, O. Spatscheck, Characterizing radio resource allocation for 3g networks, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 137–150.

[23] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, O. Spatscheck, A close examination of performance and power characteristics of 4g lte networks, in: Proceedings of the 10th international conference on Mobile systems, applications, and services, ACM, 2012, pp. 225–238.

[24] M. Zhong, P. Hu, J. Indulska, Revisited: Bandwidth estimation methods for mobile networks, in: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, IEEE, 2014, pp. 1–6.

[25] R. Ravaioli, G. Urvoy-Keller, C. Barakat, Characterizing icmp rate limitation on routers, in: 2015 IEEE International Conference on Communications (ICC), IEEE, 2015, pp. 6043–6049.

[26] A. Developers, Android network capabilities, https://developer.android.com/reference/android/net/NetworkCapabilities, [Referenced: 2020-01-18].