Large Scale "Speedtest" Experimentation in Mobile Broadband Networks

Cise Midoglu[a,*], Konstantinos Kousias[a], Özgü Alay[b,c], Andra Lutu[d], Antonios Argyriou[e], Michael Riegler[b], Carsten Griwodz[c]

[a] *Simula Research Laboratory, Norway*
[b] *Simula Metropolitan Center for Digital Engineering, Norway*
[c] *University of Oslo, Norway*
[d] *Telefonica Research, Spain*
[e] *University of Thessaly, Greece*

Abstract

Characterizing and evaluating the performance of Mobile Broadband (MBB) networks is a vital need for today's societies. Testbed-based measurements are of great significance in this context, since they allow for controlled and longitudinal experimentation. In this work, we focus on "speed" as an important Quality of Service (QoS) indicator for MBB networks, and work with MONROE-Nettest, an open source speedtest tool running as an Experiment as a Service (EaaS) on the Measuring Mobile Broadband Networks in Europe (MONROE) testbed. We conduct an extensive longitudinal measurement campaign spanning 2 countries over 2 years, and provide our experiment results together with rich metadata as an open dataset. We characterize this open dataset in detail, as well as derive insights from it regarding the impact of network context, spatio-temporal effects, roaming, and mobility on network performance. We describe our experiences about conducting speedtest measurements in MBB, and discuss the challenges associated with large scale testbed experimentation in operational MBB networks. Tackling one of the said challenges further, we introduce the notion of *adaptive* speedtest duration, and leverage a Machine Learning (ML) based algorithm to provide a proof-of-concept implementation called "Speedtest++". Finally, we describe the lessons we have learned, as well as provide an overall discussion of how open datasets can support MBB research, and comment on open challenges, in the hope that these can serve as discussion points for future work.

Keywords: EaaS, large scale measurements, mobile broadband networks, mobile testbed, speedtest, wireless communications

1. Introduction

Mobile Broadband (MBB) networks underpin numerous vital operations in today's societies, ranging from healthcare to education, transport, business communications, energy, security, and many more. Correspondingly, these networks are becoming increasingly prevalent and critical, surpassing fixed networks, and arguably becoming the most important piece of the communications infrastructure. Given the importance of MBB networks, there is a strong need to objectively assess their performance and reliability. Such assessments are valuable to multiple interested parties, including Mobile Network Operators (MNOs), content/service providers, national/international regulatory authorities and policy makers, businesses whose services depend on the underlying mobile communication infrastructures, application developers, researchers and innovators, as well as individual consumers, and the society at large.

Characterizing the performance of broadband networks in general, as well as the Quality of Service (QoS) provided by these networks to end-users, requires systematic end-to-end measurements. Several regulators have translated this need into ongoing nationwide efforts, such as the Federal Communications Commission (FCC)'s *Measuring Broadband America* initiative in the USA [1], and the *Reference Measurement System* contracted by Body of European Regulators for Electronic Communications (BEREC) to be deployed in European countries [2]. Other organizations, such as the Internet Engineering Task Force (IETF) have also published relevant documents and Requests for Comments (RFC) about large-scale measurement infrastructures [3]. These systems, however, might mainly target fixed broadband networks, and require the use of "white-box" devices that plug into cable or DSL modems to measure the speed and quality of home Internet connections.

There are multiple approaches for assessing the performance of MBB networks. MNOs can rely on passive network-side monitoring, which fails to capture the end-to-end performance due to measurement probes traditionally being located within the core network. MNOs and independent agencies can also perform drive-tests, which help identify coverage holes or performance problems. However, drive-tests are expensive, do not scale well, and also do not provide a completely end-to-end measurement.

Another approach is to rely on end-users to run performance tests directly, by visiting a website or running a special measurement application on their smartphones. The main advantage of this crowdsourced approach is scalability. Millions of measurements can be collected from different regions, networks and user device models. However, with such approaches, measurement data can only be collected at users' own will, with no possibility to either monitor or control the measurement pro-

*Corresponding author email address: cise@simula.no

cess. Due to the lack of control, such setups lack repeatability and longevity. Longitudinal measurements which reveal important information about the stability and availability of MBB networks over time, might not be possible with crowdsourced tools since they require periodic measurement sessions. Furthermore, mostly due to privacy reasons, crowdsourced measurements might not provide rich context information and metadata, such as user location, type of user equipment, Operating System (OS) build, mobile subscription, and connection mode. However, metadata is critical in putting measurement results into the right context.

These challenges have driven the research community towards testbed-based experimentation, which allows for controlled, scalable measurements over long periods of time. In this regard, Measuring Mobile Broadband Networks in Europe (MONROE) is the first hardware-based European platform for open, independent, multi-homed, and large scale monitoring/assessment of MBB performance, allowing for custom experimentation in operational networks (measurements "in the wild"), over extended periods of time [4, 5]. Although testbeds such as MONROE provide a controlled environment, the main disadvantage is scalability, as the number of measurement nodes might be limited, and much fewer compared to crowdsourced approaches.

Performance measurements in MBB often rely on speedtest tools, which can be used for many purposes such as network performance monitoring, benchmarking (over locations, network operators, technologies, etc.), bandwidth estimation for interface selection purposes (e.g., multi-path scheduling), and more. As listed in Section 2, there are a plethora of speedtest tools, provided by different interested parties. However, most of these are closed source with incompatible measurement methodologies. In order to address this challenge, we have designed and implemented MONROE-Nettest in [6]. MONROE-Nettest is an open source, Docker-based speedtest tool, which is compatible with most of the existing regulatory speedtest tools. It is built as an Experiment as a Service (EaaS) on top of the MONROE platform, but can also work standalone.

In this work, we employ the MONROE-Nettest tool on the MONROE testbed to conduct a longitudinal evaluation of end-to-end QoS in a number of prominent MBB networks in Europe. Our goal is not to compare the performance of the MNOs at scale, but rather to understand how to evaluate mobile network performance correctly. Using MONROE- Nettest, we conduct a large scale measurement campaign and quantify QoS across multiple MNOs over time and space, followed by an analysis of the impact of network context, spatio-temporal effects, roaming and mobility on "speed".

Finally, we tackle one of the most crucial challenges associated with speedtests: mobile data consumption due to the active transfer of data during measurements, which increases with increased measurement duration. To address this problem, we introduce the notion of *adaptive* speedtest duration. We leverage the Machine Learning (ML) based methodology introduced in [7] to implement *Speedtest++*, a lightweight and configurable framework capable of estimating the available data rate in considerably less time than the full MONROE-Nettest measurement duration. The tradeoff between its prediction accuracy and mobile data consumption is the basis for adaptively selecting the optimal speedtest duration.

Overall, this work is an extension and merge of [6] and [7], wherein the added novelty is that we employ the software tool described in [6] in a large scale measurement campaign, characterize and open the resulting dataset, leverage the methodology described in [7] to address one of the biggest challenges associated with large scale data collection in MBB networks, and also present the source code for the resulting proof-of-concept implementation. Our contributions can be listed as follows:

- Using MONROE-Nettest [6], we run an extensive longitudinal measurement campaign, spanning multiple MBB network configurations in 2 countries over 2 years. We provide our experiment results together with rich metadata as an open dataset.
- We characterize this dataset in detail, and derive insights from it regarding the temporal evolution of different networks, as well as the impact of network context, spatio-temporal effects, roaming and mobility on network performance. We report on our experiences about conducting speedtest measurements in MBB, and discuss the challenges associated with large scale testbed experimentation in operational MBB networks.
- We use the ML based methodology introduced in [7] on our dataset to implement *Speedtest++*, a framework for minimizing the volume of data consumption during measurements through adaptive speedtest duration. We provide the proof-of-concept implementation as open source [8].

The rest of this paper is organized as follows: In Section 2, we provide a summary of related work. In Section 3, we describe the MONROE testbed and the MONROE-Nettest tool. In Section 4, we provide an overview of the measurement setup, the experiments making up a large scale longitudinal measurement campaign, and the resulting open dataset. In Section 5, we characterize this dataset and present a list of challenges associated with large scale testbed experimentation in MBB networks. We address one of the main challenges in Section 6, and describe *Speedtest++*. We share the lessons we have learned in Section 7, followed by a discussion of further compelling applications of open MBB datasets and open challenges, which can serve as future work points, in Section 8. We conclude the paper in Section 9.

2. Related Work

In this section, we summarize relevant literature regarding the longitudinal analysis of broadband networks, speedtest tools, and the use of ML for network performance prediction.

Longitudinal analysis of broadband networks: There are a number of studies which analyse (mobile) broadband networks over a long period of time. In [9], authors use the crowdsourced tool "MobiPerf" to provide an analysis of multiple networks

over the world over 17 months, with non-controlled experimentation. Annual reports by the Center for Resilient Networks and Applications (CRNA) in Norway provide performance metrics on all Norwegian MBB networks over the course of multiple years, as measured by a designated hardware-based testbed called "NORNET" [10, 11, 12, 13, 14, 15]. Similar testbeds which allow for longitudinal experimentation in broadband networks include mLab [16] and RIPE Atlas [17]. In [18], authors investigate the state of broadband in Africa, in the form of a survey. Other studies studies such as [19] and [20] investigate the performance of MBB networks under roaming, mobility, etc.

However, none of the above studies observe multiple operational MBB networks from the perspective of dedicated, controllable and comparable clients, with regularly executed distributed measurements over the course of multiple years, with rich context information such as location, signal coverage, connection mode and mobility, as we do in this work using the MONROE platform. Our periodic measurements on comparable hardware reveal information about the stability and availability of different networks over time, as well as allow for performance benchmarking across MNOs.

"Speedtest" tools: Notable works on broadband measurements, such as [21], indicate that "speed" is the most important metric of interest in characterizing the quality of a broadband service. There are a plethora of speedtest tools, including commercial third party and operator tools such as Ookla Speedtest [22], OpenSignal [23], Zafaco Kyago [24], and TrafficMonitor, as well as regulatory tools such as RTR-Nettest [25], NKOM Nettfart [26], HAKOMetar [27], NetMetr [28], AKOS Test Net [29], and RATEL-Nettest [30], and academic tools such as MobiPerf [31], Netalyzr, and Netradar [32]. These tools have been used to varying degrees in association with research studies. In [33], authors discuss the opportunities and challenges of using crowdsourced measurements from such speedtests for mobile network benchmarking. In [34], authors employ crowdsourced RTR-Nettest measurements to investigate the characteristics of MNOs, and build a ML based framework to define and determine the behavior of different MNOs.

Most of the existing tools are closed source with incompatible measurement methodologies. Although the prevalence of Transmission Control Protocol (TCP)-based testing is documented by [21] as a common element to their methodologies, there are still many differences among the existing tools in terms of client configuration, server (network) infrastructure, and test parameters. Furthermore, as crowdsourced approaches, many of the tools are not applicable for testbed scenarios. In [6], we introduce the testbed-compatible and configurable MONROE-Nettest to address the need for large scale and longitudinal measurements over operational MBB networks. MONROE-Nettest is a highly cofigurable Docker-based implementation of a speedtest using TCP, which allows for comparable experimentation in both testbed and standalone senarios. It can also be used in conjunction with other software to investigate QoS and Quality of Experience (QoE) in (mobile) broadband, such as [35], where authors instrument MONROE-Nettest within a video streaming experiment to assist the analysis of user QoE. In this paper, we exploit this tool to run a longitudinal study of operational MBB networks.

Network performance prediction with ML: Network performance prediction is critical for a plethora of network management tasks, which include performance optimization, traffic management, application provisioning and crowdsourced benchmarking. Over the years, several attempts to model and predict performance have been proposed. Even though different studies tackle the problem from slightly different angles, they share a common goal, which is to provide high accuracy and minimize the prediction error. Experimental approaches toward performance prediction have been addressed in [36], [37], [38], [39], [40], [41] and [42]. The authors in [43] leverage constant rate probing packets for a very short duration to estimate available bandwidth in a controlled Long Term Evolution (LTE) environment while in [44] they extend their work by further testing and validating the developed framework in live LTE networks. In [45], Maier et al. introduce a novel Artificial Intelligence (AI) model that leverages feed-forward neural networks to estimate downlink and uplink bandwidth, with the aim of minimizing data consumption during speedtests. The authors make use of relatively complex neural network architectures which increase the implementation and training time complexity compared to simpler models. In [46], authors present a ML approach trying to predict the latency in an operational MBB network. Besides empirical solutions, theoretical models have also been proposed. In [47], Gao et al. introduce a theoretical learning based throughput prediction system for reactive flows, while authors in [48] propose a novel stochastic model for user throughput prediction in MBB networks that considers fast fading and user location.

While all of the above studies address the same problem, that of accurate network performance prediction, most do not pinpoint its relevance to the selection of an optimal measurement duration. We approach the problem from this specific angle: our goal is to reduce the amount of data consumed while running speedtests on MBB networks, by using high-accuracy data rate predictions. More specifically, we would like to find out the shortest possible measurement duration which is adequate for an estimation of the "true" data rate with reasonable accuracy. In Section 6 of this paper, we make use of the supervised ML based solution for data rate prediction that we have proposed in [7]. We present *Speedtest++*, a proof-of-concept implementation based on [7], which exploits the tradeoff between prediction accuracy and data consumption. *Speedtest++* allows for the dynamic configuration of several hyperparameters and algorithms. This is of utmost importance, as sensitive use cases may require a more accurate data rate estimation and consequently a higher amount of data to be transferred, whereas it is possible to opt for a relatively short duration, as low as a couple of seconds, if a rough estimation of the network capacity is adequate. We provide indicative results that can be used as a guideline by experimenters to decide upon a test duration, which offers an appropriate trade-off between measurement accuracy and data consumption for their specific purposes. Subsequently, we make the source code open for the community.

Figure 1: MONROE-Nettest system architecture.

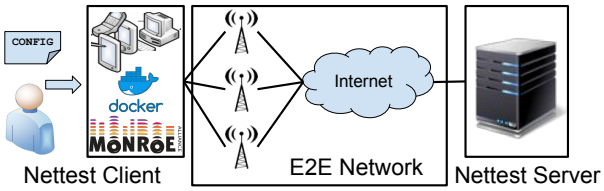
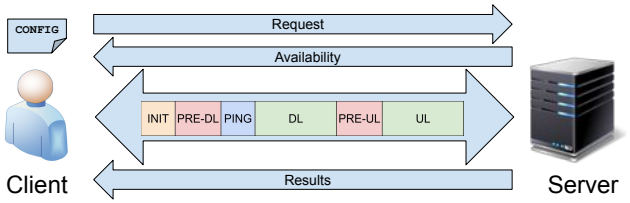


Figure 2: MONROE-Nettest application flow.



3. Infrastructure and Tool Design

In this section, we provide an overview of our experiment infrastructure and measurement tool design, detailing the configuration and usage of MONROE-Nettest.

3.1. MONROE Platform

MONROE [4, 5, 49] is a European transnational open platform, and the first open access hardware-based platform for independent, multi-homed, and large scale MBB measurements on commercial networks. The platform comprises more than 150 measurement nodes, both mobile (e.g., operating in delivery trucks and on board public transport vehicles, such as trains or busses) and stationary (e.g., volunteers hosting nodes in their homes). Nodes are multi-homed to multiple different MBB operators using commercial-grade subscriptions in several countries in Europe. Each MONROE node integrates two small programmable computers (PC Engines APU2 board interfacing with three 3G/4G MC7455 miniPCI express modems using LTE CAT6 and one WiFi modem). The software on the node is based on Debian GNU/Linux "stretch" distribution and each node collects metadata from the modems, such as carrier, technology, signal strength, GPS location and sensor data. This information is made available to the experimenters during execution. Experiments running on the platform uses Docker containers (light-weight virtualized environment) to provide agile reconfiguration. All software components used in the platform are open source and available online.

MONROE provides a controlled environment to conduct repeatable and reproducible experiments. In contrast to crowdsourced tools which cannot produce datasets for performance characterization of MNOs due to the amount of noise in their results or because of app permission requirements, MONROE provides a clean dataset collected from identical devices that require no maintenance on the part of the end user.

3.2. MONROE-Nettest Overview

MONROE-Nettest is a flexible speedtest tool built as an EaaS over the MONROE platform, which is based on and compatible with a number of European regulatory measurement tools. Below, we provide brief information on its system architecture, application flow, and configuration. More details can be found under [6, 8].

Figure 1 provides a system overview, including the possibility of running the MONROE-Nettest client on different platforms through Docker virtualization. The traffic flow between the client and the server can be accomplished through multiple (wired or wireless) interfaces, depending on availability of client hardware and configuration parameters. The connectivity between the measurement server and the Internet should ideally be accomplished through a high-bandwidth dedicated line, so that the performance of the data service under test is primarily determined by the performance of the mobile network side of the connection. [50] presents a detailed discussion on these lines, avoiding system bottlenecks, and appropriate scheduling algorithms.

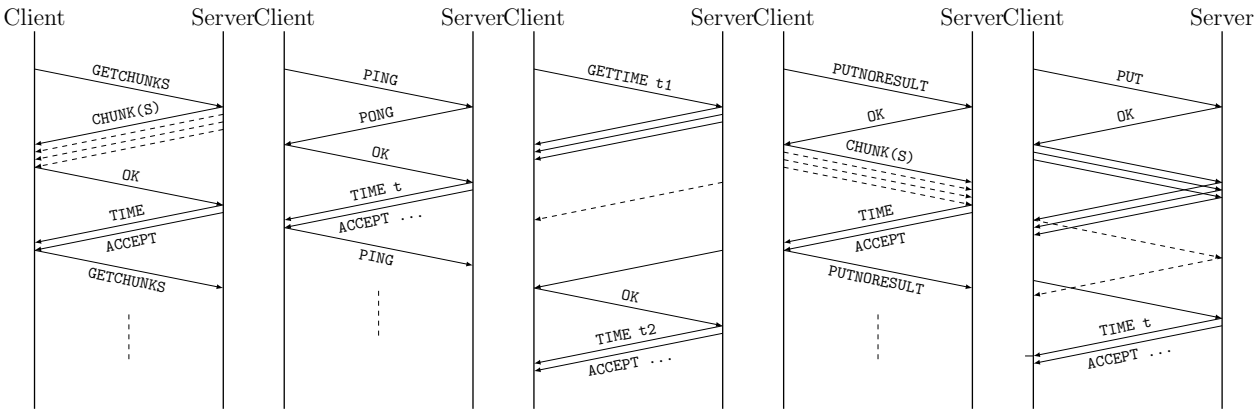
3.3. Application Flow

The MONROE-Nettest client container [51] is a wrapper around the client core, combining a number of functionalities, and making the core compatible for large scale testbed experimentation. Figure 2 depicts the application flow which is as follows: (1) the client container makes a measurement request to the server, (2) the measurement server replies with the notion of availability, (3) the measurement, composed of 6 phases, is run between the client and the server, (4) the results are gathered at the client side. This flow is easily applicable to testbeds, as well as highly scalable.

Within (3), the client container first establishes that metadata information is available, then it runs a traceroute against the selected measurement server, after which it runs the client core, and at the end, manages the output files. The main wrapper functionality has been written in Python, with a number of bash scripts to aid in the management of files.

A MONROE-Nettest core measurement consists of 6 phases: *Initialization*, *Pre-Test Downlink*, *Ping Test*, *Downlink Test*, *Pre-Test Uplink*, and *Uplink Test*. Phases are illustrated in more detail in Figure 3 with Message Sequence Charts (MSCs).

Initialization consists of the client connecting to the measurement server and establishing the desired number of TCP flows. This exchange is very brief and consists of an almost-constant number of packets. Once the client establishes a connection with the server, the pre-test DL phase follows. Pre-test phases are undertaken with the same purpose: to ensure that the Internet connection is in an "active" state, i.e. that dedicated radio resources are available. In the pre-test DL phase, for each TCP flow, the client requests data in the form of chunks that double in size for each iteration. The duration of this phase is configurable. The ping test consists of the client sending a desired number of TCP "ping"s in short intervals to the server to test the Round-Trip Time (RTT) of the connection. This exchange is also very brief and consists of an almost-constant number of packets. The number of pings are configurable. The pre-Test UL phase works analogously to the pre-test DL phase, but with the client as the sender and the server as the receiver.



(a) Pre-test Downlink (DL) (b) Ping (c) DL (d) Pre-test Uplink (UL) (e) UL

Figure 3: Message Sequence Charts (MSCs) for MONROE-Nettest measurement phases [6]

The DL and UL tests are the main components of the measurement wherewithin multiple TCP flows, the receiver side simultaneously requests and the sender side continuously sends data streams consisting of fixed-size chunks. After the nominal duration, the sender stops sending further chunks on all connections, the last chunk per each thread is allowed to transmit completely, and the DL/UL data rate of the connection is estimated.

Notes on compression and encryption: The chunks that are sent and received consist of random data to prevent any compression of the measurement data. The random data is pregenerated on the server side and the client reuses the received random data for the uplink measurement. Transport Layer Security (TLS) is used on top of the TCP streams to increase the probability that the measurement can be performed even within networks protected by firewalls and proxy servers, and to additionally prevent compression. Data security for the transmitted data per se is not a reason for using TLS. Our internal benchmarking showed no significant difference for the measurement results using TLS compared to measurements not using TLS. The cryptographic handshakes performed during TLS connection establishment are not performed during the actual measurement phase.

Data rate calculation: For the calculation of the data rate to be reported, the client uses an aggregation of all flows, with a granularity of one data chunk (which is also a configurable parameter). Let n be the number of TCP flows used for the measurement and $F := \{1, \ldots, n\}$ be the set of these flows. All transmissions start at the same time, which is denoted as relative time 0. For each TCP flow $f \in F$, the client records the relative time $t_f^{(i)}$ and the total amount $b_f^{(i)}$ of data received in Bytes on this flow (per chunk), from time 0 to $t_f^{(i)}$ for successive values of i, starting with $i := 1$ for the first chunk received. For each TCP flow $f \in F$, the time series begins with $t_f^{(0)} := 0$ and $b_f^{(0)} := 0$, where m_f is the number of pairs $\left(t_f^{(i)}, b_f^{(i)}\right)$ which have been recorded for flow f.

$$t^* := \min\left(\{t_f^{(m_f)} \mid \forall\, f \in F\}\right) \tag{1}$$

$$\forall\, f \in F : i_f := \min\left(\{i \in \mathbb{N} \mid 1 \leqslant i \leqslant m_f \wedge t_f^{(i)} \geqslant t^*\}\right) \tag{2}$$

i_f being the index of the chunk received on flow f at t^* or right after t^*. Then the amount b_f of data received over TCP flow f from time 0 to time t^* is approximately

$$b_f :\approx b_f^{(i_f-1)} + \frac{t^* - t_f^{(i_f-1)}}{t_f^{(i_f)} - t_f^{(i_f-1)}}(b_f^{(i_f)} - b_f^{(i_f-1)}) \tag{3}$$

The data rate for all TCP flows combined is given by Eq.4, where R is used as the final reported data rate.

$$R := \frac{1}{t^*} \sum_{f=1}^{n} b_f \tag{4}$$

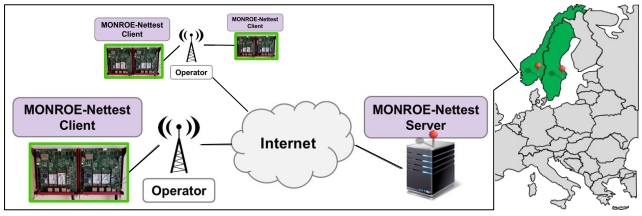
This particular calculation does not target the application data rate (e.g., including the SSL overhead if enabled), rather it measures the transport capacity by counting the bytes transmitted by the flows directly from the socket, as in Equation 4. It is possible to get both the application data rate and the transport capacity using MONROE-Nettest, where the latter can be calculated using the detailed `TCP_INFO` available from the **stats.json** output file. Also, the current calculation includes the slow-start phase of all TCP flows. It is known that some of the existing speedtest tools cut out the TCP slow start, which yields a more optimistic data rate estimate.

RTT calculation: For every TCP "ping", the RTT is calculated on both the client and server side. The client computes the difference between sending the `PING` and receiving the `PONG`, whereas the server computes the difference between sending the `PONG` and receiving the `OK` (see Figure 3(b)). Afterwards, the median of all pings are computed (therefore odd number of pings are encouraged in the client configuration), and the aggregate is provided in the summary output along with each individual result. This is referred to as the "TCP payload RTT".

3.4. Configuration

Client: The MONROE-Nettest client is highly customizable with over 20 configuration parameters, including the number of flows for DL and UL tests, measurement durations for

Figure 4: Measurement setup.



Parameter	Description	Value
`cnf_dl_pretest_duration_s`	Pre-test DL duration	$1s$
`cnf_rtt_tcp_payload_num`	Number of pings	11
`cnf_dl_duration_s`	DL test duration	$10s$
`cnf_dl_num_flows`	Number of TCP flows	5
`cnf_server_host`	Measurement server	`SE/NO`

Table 1: Configuration parameters employed throughout the measurement campaign.

DL, UL and pre-test phases, number of pings, and measurement server (hostname and port). It is also possible to use the `multi_config` parameter, which enables setting groups of configuration parameters together, to be executed in batches. The configuration parameters can be passed to the Docker container as a JavaScript Object Notation (JSON)-formatted string.

Server: Configurable parameters of the server include the port numbers, and the allowed number of parallel client connections. The configuration parameters have to be set while the server code is being compiled.

The full list of MONROE-Nettest configuration parameters and default values can be found under [8]. In Section 4.1, we list the parameters we explicitly set for our measurement campaign on the client side.

4. Measurement Setup, Experiments and Dataset

In this section, we describe the longitudinal measurement campaign we have been running using MONROE-Nettest, as well as the corresponding massive dataset we are making publicly available with this work as open data.

4.1. Experimental Setup

Figure 4 illustrates our measurement setup. As **clients**, we have employed 24 stationary nodes in Norway, 36 stationary nodes in Sweden, 31 nodes on trains in Norway, and 48 nodes on buses in Sweden. As **servers**, we have employed 2 well-provisioned MONROE-Nettest servers in Norway and Sweden, deployed on virtual machines hosted by the MONROE Alliance, which are indicated as `NO` and `SE` in Table 1. The idea for employing multiple measurement servers in different countries is to make sure that during measurements, a client node connects to the server towards which it has the lowest network latency. Geographical vicinity has been shown to have a negative correlation with network latency [21], therefore each node targets the physically closest server.

We have scheduled MONROE-Nettest as a *base* experiment on all available nodes in Norway and Sweden for over 2 years, starting from 2018 (see Section 4.3 for notes on availability). This corresponds to running the client container with the fixed configurations provided in Table 1, two or more times a day on all available nodes. As mentioned above, client nodes in Sweden (including mobile nodes) have been configured to run measurements against the server in Sweden, where client nodes in Norway (including mobile nodes) have been configured to run measurements against the server in Norway.

We have exported MONROE-Nettest base experiment results, as well as the metadata stream of interest (`MONROE.META.DEVICE.MODEM`) corresponding to the times of the experiments, from the MONROE repository using a consortium experimenter certificate. We process this raw data and present it as an open dataset, as described in Section 4.2.

4.2. Dataset

In order to generate our longitudinal MBB dataset, we have used a part of the raw data from MONROE-Nettest measurements between 01.01.2018 and 31.12.2019, along with experiment metadata from the MONROE platform, to produce 2 types of files as described below.

Summary results per day: Daily aggregate `monroe-nettest-<yyyy-mm-dd>.CSV` files combine the configuration parameters, reported performance metrics (DL data rate, UL data rate, RTT), and corresponding metadata (Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), mobile technology, and mobility indicator) for each MONROE-Nettest measurement within a given day day, and include one line per measurement.

Detailed results per measurement: The `<measurement-id>.JSON` files per each MONROE-Nettest measurement provide all the information pertaining to the measurement in the corresponding daily aggregate file specified above, as well as detailed time series for all phases (e.g., timestamped records of downloaded bytes per chunk in the DL phase). One `JSON` file is generated for each measurement, and files are sorted into daily folders.

We provide these `CSV` and `JSON` files as a compressed archive under [8]. Overall, our dataset contains over $150K$ MONROE-Nettest measurements from 6 SIM and network operators. Table 2 lists the measurement statistics.

4.3. Technical Limitations

The limitations of our dataset in terms of hardware, spatio-temporal coverage, and mobility are addressed below.

Hardware: Since all deployed MONROE nodes integrate `MC7455` modems, our measurements only consist of a single type of device category (`LTE CAT6`). As a fixed hardware setup, this was not possible to modify during our measurements. `LTE CAT6`, as a relatively non-recent device category, has limited representativity over next generation mobile devices. However, this property of our dataset provides an invaluable opportunity to investigate the QoS as experienced by legacy devices in today's networks (see Section 5).

Scenario		2018	2019
Native	NO-op1	12370	7342
	NO-op2	13115	8783
	NO-op3	7173	6537
	SE-op1	24596	12217
	SE-op2	24053	12641
	SE-op3	22035	12142
Roaming	SE-op1 in NO-op2	3003	250
	NO-op3 in NO-op2	0	244
	NO-op1 in SE-op3	26	8
Total	Stationary	70994	34280
	Mobile	35386	25895

Table 2: Number of measurements in native scenarios (top), roaming scenarios (middle), and in total on stationary and mobile nodes (bottom).

Scale: Since our experiments rely on the MONROE platform as the underlying infrastructure, our measurements are limited to the physical setup and scale of its deployment. As of June 2020, the testbed geographically covers 4 European countries (Italy, Norway, Spain, Sweden), with a total of 24 nodes available for development and 173 nodes available for experimentation. Among these, 38 are deployed on trains and buses. Roaming is enabled on all SIMs, but it is not possible to *control* when this occurs. The middle section of Table 2 shows the uneven number of measurements performed in roaming for the top 3 operator pairs (in terms of measurement count).

SIM quotas: As a very large platform providing end-to-end measurements in *operational* MBB networks, MONROE is realistically limited to commercial SIM subscriptions. These often incur monthly quotas on the volume of mobile data that can be consumed. This has affected the temporal and spatial scale of our measurements in the following ways: (1) we have scheduled the MONROE-Nettest *base* experiment only 2 (and later 3) times per day on the nodes, (2) we have not been able to schedule as many experiments in Italy and Spain as we have in Norway and Sweden since the monthly quotas from MNOs in Italy and Spain were lower, measurements from these countries were therefore omitted in this study. We have tried to compensate for these limitations by the sheer longitude of our campaign (2.5 years), as well as by using nodes from all available locations in Norway and Sweden, in order to make sure the measurements are geographically well dispersed despite being in the same country.

Mobility: The main assumption in our analysis of the "mobile" nodes was to consider these nodes as constantly mobile, which later transpired to be false. A preliminary visualization showed that the vehicles on which the nodes were deployed could at times, and quite often, be stationary (either due to mandatory maintenance in the garage, or simply having idle times between trips). In this regard, the MONROE metadata, which also includes real-time location measurements from the external GPS antennas associated with the nodes, has played a very important role. We have filtered out MONROE-Nettest measurements which were started while the "mobile" nodes were stationary, in our analysis.

5. Open Dataset Characterization and Analysis

In this section, we highlight the main properties of the dataset that was described in Section 4.2, while using its longitudinal characteristics to make some observations and derive insights about the state of a number of prominent MBB networks in Europe.

We begin by looking at network context, which is a representation of the physical characteristics of the underlying communications infrastructure. We derive insights about the effect of mobile technology, network operator, and signal coverage on reported network performance in Section 5.1. Next, we focus on spatio-temporal features, which represent user behaviour. We investigate the impact of location, time of day and day of week on network performance in Section 5.2. Finally, in Section 5.3, we consider the impact of roaming (national and international) and mobility on network performance.

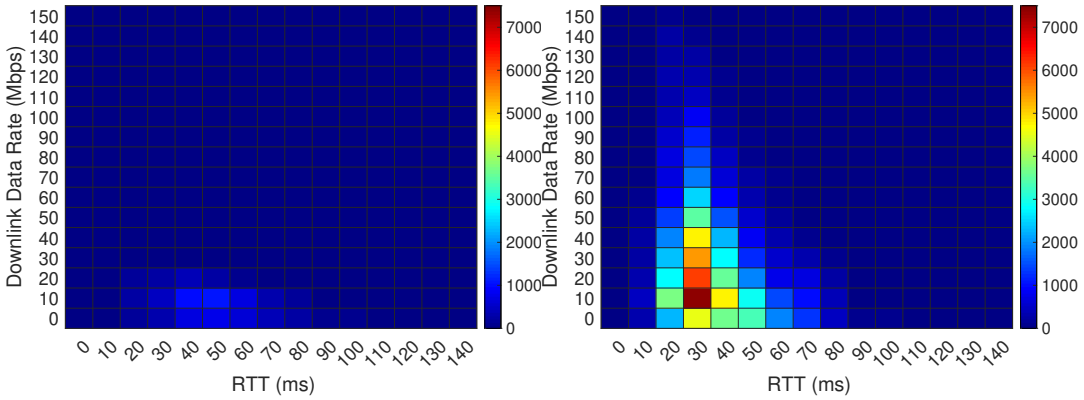
5.1. Network Context

Impact of mobile technology: We start by investigating the impact of mobile technology on network performance in terms of DL data rate and RTT. For this analysis, we consider $44K$ stationary and non-roaming measurements that were performed in 2018 and 2019. Among these measurements, $39K$ have been conducted in 4G, and $5K$ have been conducted in 3G.

Figure 5 presents the distribution of all samples with respect to the two performance metrics, grouped according to mobile technology (with no distinction with respect to network). We observe that 3G and 4G measurements have a different range for both performance metrics as expected, with the majority of 4G measurements distributed around $38ms$ RTT - $10Mbps$ DL data rate, and 3G measurements distributed around $51ms$ RTT - $7Mbps$ DL data rate. The difference between 3G and 4G access technologies are irrespective of MNO, i.e., all operators show more or less the same characteristics in terms of the difference between 3G and 4G.

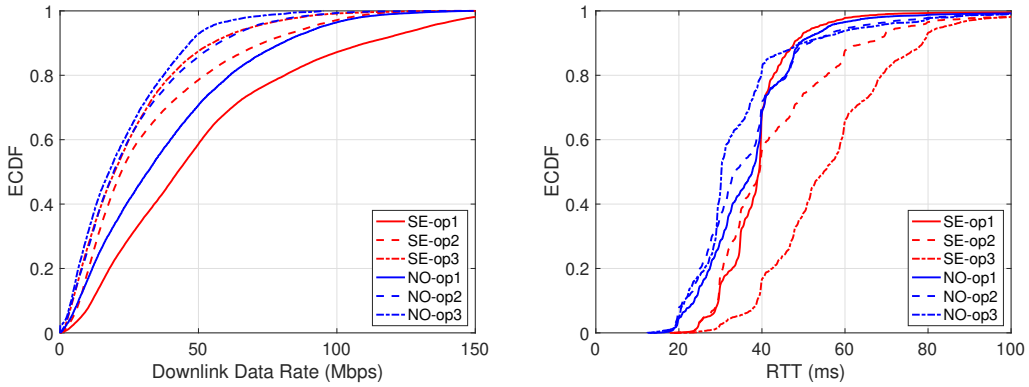
Impact of network operator: Next, we look at a number of selected MNOs, and characterize their performance in native (non-roaming) 4G scenarios. For this analysis, we consider DL and ping measurements run in 3 MNOs from Norway and Sweden each. Figure 6 presents the Empirical Cumulative Distribution Function (ECDF) of DL data rate and RTT per MNO. We observe that there is a difference in performance between different operators, although they might be from the same country/region. Overall, MNOs in Sweden operate with higher median DL data rate ($43Mbps$ for `SE-op1`, $22Mbps$ for `SE-op2`, $20Mbps$ for `SE-op3`), but they also have a higher median RTT ($40ms$ for `SE-op1`, $39ms$ for `SE-op2`, $55ms$ for `SE-op3`).

Impact of signal coverage: In order to observe the effect of signal coverage on DL data rate, we consider the reported DL data rate value from each MONROE-Nettest measurement, together with the last signal strength value reported by the corresponding MONROE client node interface before the start of the measurement. As the signal strength metric, we use RSSI for measurements conducted in 3G, and RSRP for measurements conducted in 4G. Figure 7 presents the distribution of DL data rate with respect to signal coverage in the form of a



(a) 3G (b) 4G

Figure 5: Heatmap of the distribution of performance in terms of downlink data rate and round-trip time, per mobile access technology.



(a) (b)

Figure 6: Performance in terms of downlink data rate and round-trip time, per operator.

scatter plot, jointly for all operators, per mobile access technology. The smoothed mean and max envelopes indicated on the individual subfigures help us observe that there is a more prominent positive correlation between signal strength and data rate in 4G, as compared to 3G.

Takeaways: Overall, network context plays a significant role in determining performance, as measured by DL data rate and RTT. Network performance is impacted most by the mobile access technology (with 4G performing better than 3G in terms of both metrics) and signal coverage (with higher signal strength values corresponding to higher performance overall). We also observe that different MNOs have different data rate and RTT profiles, which we are able to express quantifiably. Our dataset allows for making detailed profiles of data rate vs. signal strength per MNO as well, similar to what is demonstrated in Figure 7.

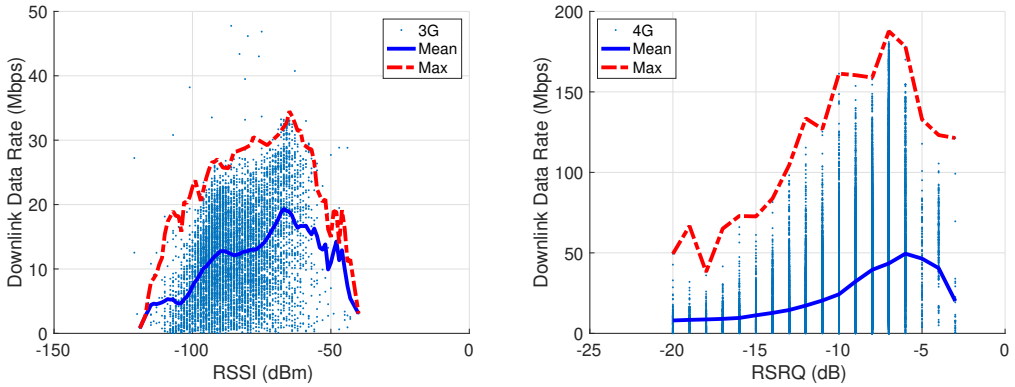
5.2. Spatio-Temporal Effects

Impact of location: Next, we investigate the impact of spatio-temporal features on the performance of MBB networks, using the stationary nodes. When we look at the performance in terms of DL data rate with respect to the country of the client, we see a trend similar to what is demonstrated by the MNOs. Figure 10(a) shows that nodes originating from different countries (equipped with SIM cards from the MNOs native to the origin country) show different performance, in accordance with Figure 7. The median DL data rate for nodes in Norway and Sweden are $24 Mbps$ and $33 Mbps$ respectively.

Impact of day-of-week: Figure 8 shows the average reported DL data rate per day of the week, for 6 operators. We see a general trend of increased data rates during Friday and Saturday, compared to the rest of the week. The overall "weekday" (Sunday-Thursday) average is $51.21 Mbps$ whereas the overall "weekend" (Friday-Saturday) average is $53.72 Mbps$.

Impact of time-of-day: Figure 10(b) shows the average reported DL data rate as reported per time-of-day (hourly granularity), for 3 MNOs in Sweden. We see a general trend of increased data rates during the night hours, and decreased data rates during working, especially rush and evening hours (average DL data rate for 0-8am: $37 Mbps$, 8am-16pm: $31 Mbps$, 16pm-0am: $30 Mbps$). The same trend is displayed by the MNOs in Norway as well.

Takeaways: The impact of location is derivative of the impact of MNO, in that, for non-roaming stationary scenarios,



(a) 3G (b) 4G

Figure 7: Performance in terms of downlink data rate, with respect to signal coverage, peer mobile access technology.

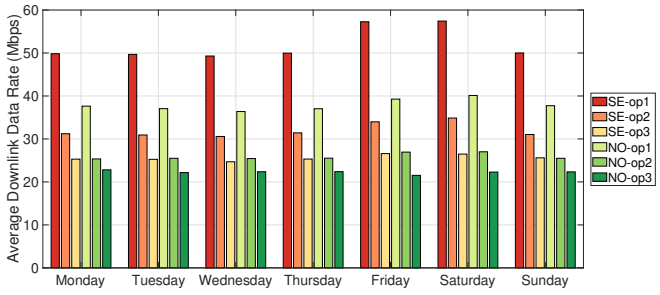
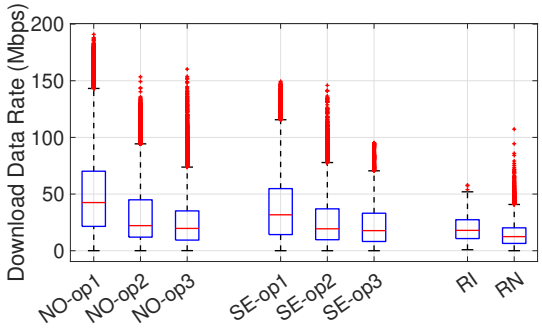


Figure 8: Average downlink data rate per day of week for 6 mobile network operators in Norway and Sweden.

Figure 9: Downlink data rate for native and roaming scenarios (RI: international roaming, RN: national roaming).



performance curves closely follow the aggregate performance of individual operators from each country. Temporality has a significant impact of network performance, most clearly visible in terms of diurnal patterns (periodic performance fluctuations with respect to time-of-day), but also as a less prominent weekly pattern, for all networks. In a broader temporal scale, over the course of multiple years, we do not observe a major change in the performance of different networks or MNOs. It is possible to conclude that the performance of existing 3G and 4G networks have converged to a relatively stabilized maximum with respect to infrastructure.

5.3. Roaming and Mobility

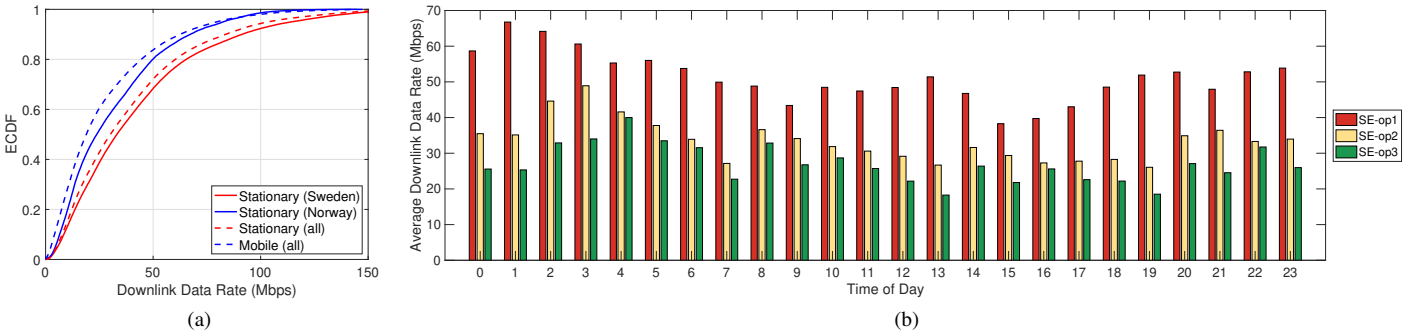
Impact of roaming: Figure 9 shows the distribution of DL data rate for 8 different operator combinations. On the left and middle, we see 6 SIM operators in Norway and Sweden which are using their own access network. On the right, we see (RI) an international roaming scenario where the SIM operator `SE-op1` is using the access network of `NO-op2` (i.e., a Swedish operator is roaming in a Norwegian operator's network), and (RN) a national roaming scenario where the SIM operator `NO-op3` is using the access network of `NO-op2` (i.e., a Norwegian operator roaming in another Norwegian operator's network). It can be observed that the median data rate tends to decrease from native to national roaming to international roaming scenarios.

Impact of mobility: Finally, we look at the reported performance in terms of DL data rate, with respect to the mobility of corresponding client nodes. After verifying that the measurements from the mobile nodes are actually executed in mobility, we group all measurements into 2 categories: `stationary` representing stationary nodes, `mobile` representing mobile nodes deployed on intra-city buses in Sweden and on high speed trains in Norway. Figure 10(a) presents the ECDF of reported DL data rate with respect to different mobility categories. We observe that the difference between reported DL data rate from stationary and mobile nodes is around $6Mbps$ in the median (50%-ile) range, and up to $9Mbps$ in the 75%-ile range, in favour of stationary nodes.

Takeaways: Roaming clients perform worse than native clients on average, with no significant difference between national and international roaming. Mobile clients perform slightly worse than stationary clients, however the level of mobility, and mobility itself has a minor impact.

5.4. Challenges

During the course of designing, implementing, scheduling and running experiments with, as well as collecting data and analysing results from MONROE-Nettest, we have had valuable experiences in tackling challenges related to next generation networks, testbed-experimentation, and TCP-based testing.



(a) (b)

Figure 10: (a) ECDF of downlink data rate per country (solid lines), and per mobility (dashed lines). (b) Average downlink data rate per time of day for 3 mobile network operators in Sweden.

In this section, we elaborate on some of these challenges which have guided our next steps.

Increased data rates: As mentioned in Section 4.3, one of the biggest challenges of operating a mobile testbed is the large cost associated with maintaining numerous SIM contracts. Limited contracts provide inaccurate data rate estimations due to the mechanisms employed by network operators to implement tariff bottlenecks, as well as completely hinder the possibility to schedule experiments after the quotas run out. This data consumption challenge exacerbated by the ever increasing speed of MBB. The ever-growing mobile data traffic will maintain its upward trajectory as we enter the Fifth Generation (5G) and Internet of Things (IoT) era.

According to the Cisco Global Mobile Data Traffic Forecast [52], the number of global mobile devices and connections is set to reach a staggering 13.1B by 2023. In addition, the average 5G speed will be 575 Mbps. The increasing data rates lead to increased data volume consumption for all active measurement based speedtests, proportional to measurement duration. As a consequence, there is a strong need to design, implement and deliver solutions that aim to restrain the excess consumption of data during speedtesting, while preserving the desired level of accuracy.

Increased use of MBB networks in mobility: Customers' ever increasing demand from MBB networks for availability under mobility (e.g., working and/or streaming content on trains, cars, buses, etc.) creates a need for conducting reliable network measurements under mobility as well. Under high speeds, long test durations cause lower accuracy and complications in terms of mapping performance metrics to specific points on the map (difficulty of mapping a reported data rate or RTT value to a single location point on a route). The longer a measurement takes, the harder it is to pinpoint which network infrastructures along the vehicle's designated route (e.g., which cells in the vicinity of train tracks) are used, and the harder it is to associate a certain performance metric with a certain route segment. Therefore, it is imperative to obtain measurement results quicker, especially in mobility scenarios. A shorter measurement duration can alleviate this challenge, and allow for performance metrics to be more easily associated with route segments.

TCP slow start: The use of TCP for measurements can be disputed. Due to TCP's slow start and congestion avoidance algorithms, it can require a considerable number of RTT's, sending an increasing packets per RTT until it reaches the equilibrium that is used as the throughput estimate. We pay for this probing period in data quota. User Datagram Protocol (UDP) does not suffer from TCP's need send data for several round-trip times until it can settle for a suitable data rate. It could probe more aggressively or start to probe with a data rate that was reached before. Mobile nodes and servers in our measurement infrastructure are known, and UDP is not stopped by firewalls. This could reduce the consumed data quota. However, UDP probing using our own rate adaptation algorithm [53] would influence competing TCP flows in different ways than a TCP probe. By choosing exteme aggressiveness, UDP could be used to probe network capacity instead of the throughput that is achievable by a typical application. We did therefore select TCP-based probing to maximize realism, and reduce the consumption of data quota by investigating when probing can be terminated with a moderate loss of accuracy in throughput estimation. Our estimation of throughput is representative of how applications "see the network", in the sense that most applications today rely on TCP as the transport protocol.

Next steps: In the following, we attempt to create a speedtest with *adaptive* duration. Our solution aims to alleviate the challenge of increased data rates and address the need for quicker measurements, by decreasing the measurement duration while preserving accuracy. We achieve this by by taking into consideration the characteristics of data rate time series, and optimize our measurement algorithm according to the trade-off between mobile data volume consumption and accuracy.

6. Speedtest++: Toward Adaptive Speedtest Duration

As mentioned in the previous section, one of the biggest challenges associated with running active measurements in operational MBB networks is the huge volume of data consumed. On the one hand, network operators face with traffic profiles of higher requirements and can be forced to adapt their network policies and resources to provide their subscribers with

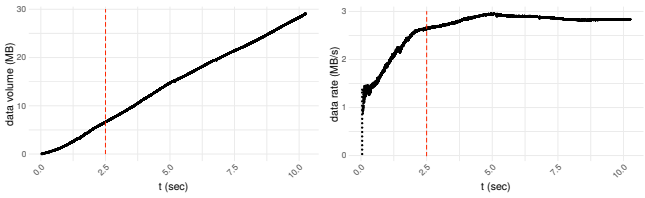
the necessary service. On the other hand, end-users with limited SIM mobile contracts and high tariffs are especially threatened. In this section we present *Speedtest++*, an open source and lightweight ML based framework that allows for *adaptive* speedtest duration toward reducing the consumed data volume over a network connection. In a nutshell, *Speedtest++* offers a solution to accurately predict a network's capacity by exploiting *passive* data (i.e., extracting network information without injecting additional traffic) and a significantly smaller portion of *active* (i.e., measuring performance by initiating data exchange over the connection) TCP traffic. Adapting the speedtest duration value is tightly coupled with the learning model's predictive accuracy. A good performing model is critical for reducing the duration of the transmission process, and therefore, prevent excess data volume consumption. We evaluate its performance by drawing 2 years of data from our dataset introduced in Section 4.2. The main results reveal a critical tradeoff between prediction accuracy and data consumption. We argue that the optimum speedtest duration value strongly relates to the application's error sensitivity requirements.

6.1. The Big Evil: Excess Data Volume Consumption

As discussed in Section 1, there exists a plethora of speedtest solutions available on the market, with plenty of them being commonly used by a significant share of subscribers who intent to access the characteristics of their own network connection. What is not instinctive, however, is that speed monitoring can actually be costing a significant portion of cellular data. For example, during a $t = 10$ seconds speedtest, the network will be flooded with data streams until it reaches its maximum bandwidth capacity and transition to what it is called the *saturation* phase. Figure 11a depicts the transmitted data (in MB) for a sample speedtest as a function of time [1]. During the early stages, we observe a nonlinear behavior between the two variables as an effect of the TCP slow start phase. However, after exceeding the slow start threshold, the curve begins converging to the identity line, hence, signaling that the channel reached its maximum capacity. We highlight the importance of selecting an appropriate speedtest duration value by drawing a vertical line that mark the end of the speedtest at a $t* < 10$ seconds. It is evident that a shorter speedtest duration will preserve a significant portion of the data, while maintaining the accuracy of the data rate estimate at sufficiently high levels. The pattern of evolution is indicative and may differ in some extent between speedtests due to the unpredictable nature of the wireless medium in MBB networks.

In the following, we present the system design of *Speedtest++* as we strive to predict the data rate that would have been reported during a speedtest, within a duration shorter than the default 10 seconds. However, *Speedtest++ is designed in a modular fashion so that it can easily be configured and validated against any measurement duration value, which might be of interest since these vary across existing speedtest tools.*

[1] The data rate (in MB/s) counterpart can be also seen in Figure 11b.



(a) Data volume (in MB) (b) Data Rate (in MB/s)

Figure 11: Time series evolution of a speedtest example in terms of data volume and data rate.

6.2. Time Series Analysis

Speedtest++ is a lightweight *R*-based ML framework that allows for data rate prediction under dynamic duration scenarios. We make the source code available to the community to accommodate reproducibility but also allow for software modifications to further boost the prediction accuracy. *Speedtest++* consists of three main methodology blocks, i.e., *feature engineering*, responsible for data cleaning, merging and transformation, *feature selection*, for improving accuracy and reducing training time and probability of overfitting, and *model design*, for training and testing our predictive models. We evaluate *Speedtest++* in terms of prediction accuracy and data volume consumption.

6.2.1. Feature Engineering

First, we aim to establish a systematic methodology for expressing each time series with a smaller number of features. Raw data (i.e., 5 TCP flows per speedtest) are available in our dataset (**flows.json**) with a granularity of 100 ms. *Speedtest++* utilizes, first, a linear interpolation scheme to perform partial sub-sampling, and second, an algorithm that leverages first degree polynomial curve fitting to compress each sequence and express it by two engineered features.

Linear interpolation: In MONROE-Nettest, time series granularity is a hyperparameter set by default to 100 ms. However, its precision may be somewhat affected by *measurement noise*. To overcome this challenge, we apply a linear interpolation method to partially sub-sample each sequence in equal spaced *fragments*, i.e., we repeatedly generate an artificial data point within the range of two known data points[2].

First degree polynomial curve fitting: To reduce training time and add simplicity to our models, we map each sequence to a couple of regression coefficients by using first degree polynomial curve fitting. The proposed algorithm follows the steps below. We first congregate the available fragments in *clusters* of size five. Using input data from the first cluster, we next apply linear curve fitting to capture the relationship between the response (bytes) and the explanatory variable (time). The coefficients `beta0` and `beta1` are known as *intercept* and *slope*. At

[2] Since speedtest duration is 10s, each sequence consists of 100 equidistant fragments. Sequence up-sampling can be also applied, however, we argue that it adds to the preprocessing time while it can likely cause overfitting.

the next iteration, we linearly concatenate the adjacent cluster to form a double in size that consists of 10 fragments. Likewise, we fit a linear model and we obtain a new pair of coefficients. We repeatedly proceed until we merge the entire sequence. We visualize the first three iteration of our algorithm in Figure 12. Note that at each iteration, the updated line fit approaches the ground-truth.

Let n represent the number of data points in a single cluster, then the first degree polynomial curve fitting equation is expressed as $Y_i = \beta_1 X_i + \beta_0 + \epsilon_i$, where Y_i, $i \in \mathbb{Z} : i \in [1, n]$ is the transmitted bytes and X_i is time. The coefficients (β_1) shows the magnitude of the effect that the explanatory variable has on the response variable given that the remainder of explanatory variables remain constant (if any). The sign signifies whether this effect is positive or negative. Last, β_0 stands for the intercept term while ϵ_i is the prediction error.

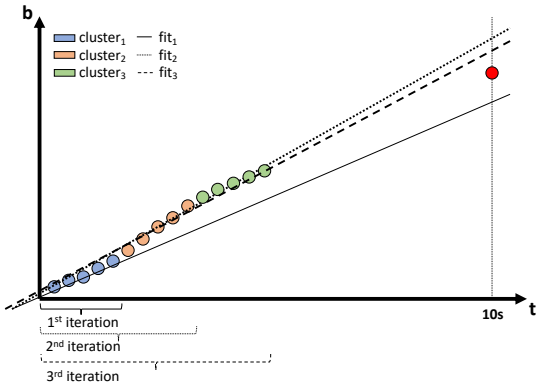


Figure 12: Visualization example of the first three curve fitting iterations using data point artifacts. x-axis represents time while y-axis is the accumulated data transferred throughout the course of the speedtest. Each of the colors maps to a single *cluster* while the line shape dictates the updated line fit at each iteration.

Discussion: We design the *feature engineering* block in a modular fashion so that it allows for experimentation with different hyperparameters, such as the number of fragments and clusters. In addition, there is enough room for exploiting different interpolation schemes (e.g., polynomial, splines) or different curve fitting equations (e.g., second or higher degree).

6.2.2. Feature Selection

Next, we apply feature selection to select the most relevant features for our predictive models. Among the available options, we select *forward selection*, a data-driven iterative process that leverages a model fit criterion to decide on the importance of the available explanatory variables. In forward selection, the starting model (known as the *null* model) has zero features. At each iteration the most important feature is added to the model until no further improvement is obtained. Examples of stopping metrics include p-values, adjusted R-squared, Akaike Information Criterion (AIC), and so forth. In this work, we use AIC, i.e., defined as $AIC = 2k - 2ln(L)$, where k represents the number of available features while L is the maximum value of the likelihood function of the model.

Data collection can incur specific *cost* in terms of data volume required for collecting specific network attributes. In Table 3 we label each available feature as either passive ($ID \in [1, 11]$) or active ($ID \in [12, 13]$). As a result of this grouping, we slightly modify the forward selection algorithm so it is performed in two stages. First, we only consider the passive features as available candidates for selection. When no improvement is observed, we update the list with the active ones. To quantify the error diversity, we show results for five different duration scenarios ($t \in [1, 5]$s). We divide Figure 13 in two parts, where each part consists of the passive and active features, respectively. x-axis shows the ordering of the features at each iteration of the forward selection while y-axis shows the corresponding AIC. All five duration scenarios are represented by a different color as sketched in the legend. Note that for the passive part, AIC score across different duration values remains unchanged. This is due to the fact that passive features do not hold any dependencies with the time domain. We observe a substantial improvement when adding β_0 and β_1, clearly revealing that knowledge of data volume patterns even at the early stages of a speedtest provides significant gains to the learning model. For higher duration scenarios, we see that AIC drops even more rapidly, which is quite intuitive, since additional data volume related information is incorporated in the model.

ID	Feature	Short Description
1	`ping_med`	RTT average (ms)
2	`ping_std`	RTT standard deviation
3	`rsrp`	RSRP (dBm)
4	`rsrq`	RSRQ (dB)
5	`rssi`	RSSI (dB)
6	`day`	Day of the week
7	`time`	Hour of the day
8	`wknd`	Weekend indicator
9	`server`	Measurement server
10	`net`	Network operator
11	`mobility`	Mobility indicator
12	`beta_0`	Intercept
13	`beta_1`	Slope

Table 3: Parameters in feature selection stage.

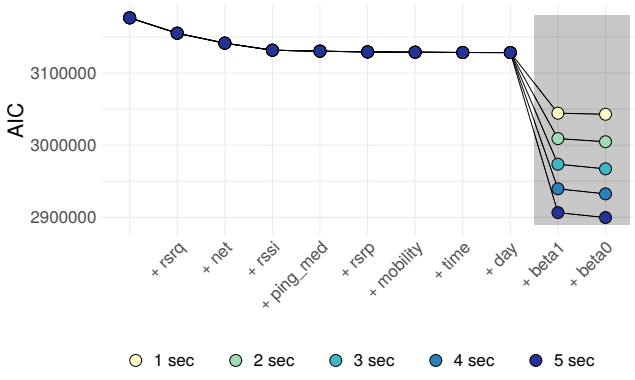


Figure 13: Forward selection for $t \in [1 - 5]$: Features in the x-axis are listed with the same order as they are added to the model during forward selection. The y-axis represents the respective AIC. The grey grid divides the downlink beta coefficients from the pool of passive features.

Discussion: A list of alternative feature selection approaches include the backward, bidirectional and recursive feature elimination. In addition, well established embedded methods, such as the lasso or ridge regression, support inbuilt penalization functions to reduce overfitting[3]. Last, feature importance can be determined by using entropy-based solutions like information gain. *Speedtest++* can be easily modified to provide support for any of the above. However, results may be slightly affected as it is likely that some algorithms may provide a different view of the most critical features.

6.2.3. Model Selection

The last piece of the framework is model selection, where we leverage the power of ML to train, validate and test numerous supervised predictive models. Our goal is to estimate the number of transmitted bytes at each of the selected duration values using the features decided in the previous stage. We consider Multiple Linear Regression (MLR), a simple but rather efficient algorithm that has found application in a variety of scientific fields including business, economics and medicine. To approximate the optimal solution, MLR leverages the linear least squares fitting approach that minimizes the sum of squares between the predicted and the groundtruth data [54]. The formal mathematical expression of a MLR model is a generalized form of the equation we introduced in the feature engineering subsection. Let f represent the number of available explanatory variables, then b_1 and X_i can be altered with b_i and $X_{j,i}$ respectively, where $j \in \mathbb{Z} : j \in [1, f]$ and $i \in \mathbb{Z} : i \in [1, n]$.

Discussion: In a similar fashion, *Speedtest++* allows for experimentation with a plethora of ML algorithms, from Support Vector Regression (SVR) and Random Forests (RF) [7], to more advanced deep learning solutions, that bring neural networks and a variety of artificial intelligence elements into play.

6.3. Performance Evaluation

We organize the following section in three main parts. First, we provide a brief overview of the hardware and we describe the ML configuration settings. Next, we present our main findings by focusing on the tradeoff between predictive error and data volume consumption under different duration values. Finally, we provide a discussion regarding model training and accuracy.

Experimental design: *Hardware* – To accelerate data preprocessing and training time, we conduct all experiments in a x86-64 architecture machine with 16GB RAM and a multithread CPU featuring 18 cores[4]. *ML Configuration Settings* – We extract a subset of the dataset introduced in Section 4.2, featuring a time period of 2 years (2018-2019). This subset consists of 2.7M samples that we split into training (67%) and testing (33%) data by using systematic sampling. We remove invalid or false measurements that can affect the reliability of our results (e.g. samples with negative values for data rate or RTT, missing RSRP or outside the valid range and so forth). However, we do not treat extreme but rational values as outliers since they can explain signs of variation in the data. To quantify and compare between different duration values, we select the Median Absolute Percentage Error (MdAPE) as our error metric. Finally, to provide agility and robustness, we adopt repeated cross validation during the training process with 10 folds and 3 repeats, which is a common configuration when comparing ML models. Furthermore, since we do not have any baseline to compare our models against, cross validation serves as a validation tool that ensures a stable and generalizable solution.

Results: Figure 14a illustrates the percentage of data consumption when using passive or active monitoring respectively. Data consumption is determined as the number of bytes exchanged until a nominal duration divided by the total amount of bytes they would have been exchanged if the speedtest was run for the default 10 sec[5]. For all passive features data consumption percentage always equals to 0%. Furthermore, we observe that data volume is almost equally distributed throughout the 10 seconds time window. Likewise, Figure 14b depicts the reciprocal MdAPE values for each of the forward selection iterations. Again, we observe that the addition of the beta coefficients highly improves the performance and significantly contributes in reducing the MdAPE. Furthermore, we find that higher speedtest duration values improve the accuracy levels in an analogous manner. For example, the MdAPE decrease between the 1 and 2 sec transmission duration is 6.71% in average which is pretty significant for highly sensitive applications.

To complement our analysis, we further train a MLR model with downlink as the dependent variable and all the features selected during forward selection as the explanatory variables. This step aims to divulge the impact of higher speedtest duration times on the estimation error. Figure 15 depicts error boxplots along different duration values where each cluster represents a $500ms$ increase in the speedtest duration. We also overlay the respective data consumption percentages for each of the clusters. We observe that, consistent with our expectations, both variance and MdAPE follow a decreasing trend for higher duration values (i.e., the longer the measurement, the closer our prediction comes to the groundtruth), while, data volume consumption linearly increases with time.

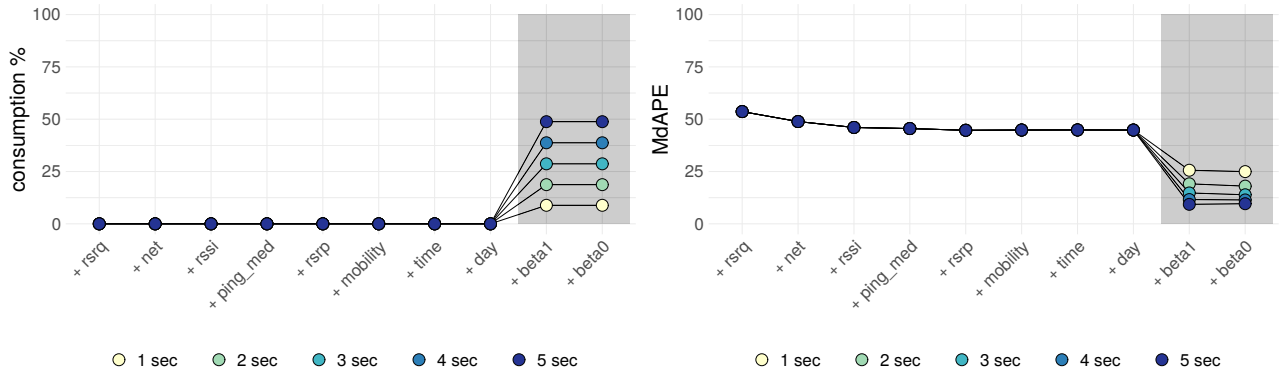
Discussion: First we refer to the model validity and discuss how vital model retraining is for maintaining accuracy in satisfactory levels. Second, we comment on the optimal speedtest duration value that should be used in *Speedtest++*.

In general, model retraining is required when recent data come from a distribution different than the one the model was originally trained from. There are three alternatives that can be used for retraining, i.e., online, offline, or by using a batch based approach. Hence, how often should we retrain *Speedtest++*? If there is no change in the MONROE-Nettest methodology,

[3] In the same category, tree-based algorithms, such as random forests, use the notion of *gini impurity* to perform feature ranking, thus proving extremely efficient.

[4] Our server is based on a Linux Ubuntu 16.04.6 LTS distribution.

[5] Within the scope of this work, we consider the default MONROE-Nettest speedtest duration, i.e., $10s$, as our ground truth value. However, different tools recommend different values, hence, models need to be updated to address these changes.



(a) Median data consumption percentage

(b) MdAPE

Figure 14: Tradeoff between predictive error and data volume consumption. The order of the network features as selected during forward selection is illustrated in the x-axis.

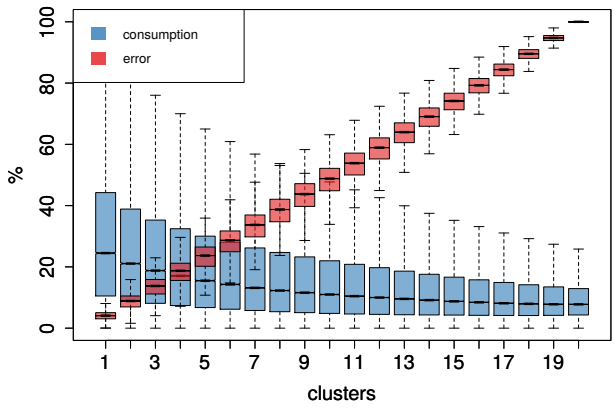


Figure 15: Absolute error and consumption percentages along different duration values illustrated in a boxplot fashion. Outliers have been removed to increase readability.

model retraining is not critically required. Such an action would only marginally decrease the predictive error, e.g., in scenarios where there are seasonal trends that haven't been captured in the previous training dataset. However, if MONROE-Nettest undergoes certain modifications or even replaced by a new tool, retraining is almost imperative as *Speedtest++* would not be able to detect new patterns in the most recent data, leading to a significant accuracy degradation. We plan to run a detailed analysis of the horizon of the model predictions as part of future work.

Regarding the optimal speedtest duration value, we find that there is no easy answer, since it is highly related to the application's error sensitivity requirements. For example, for applications that require a rough estimation of the network bandwidth capacity without having to consume a lot of data traffic, a duration value shorter than a couple of seconds is adequate. However, for applications that require high precision, a longer duration is required, though inevitably increasing data volume consumption. We recommend that readers use the two subplots in Figure 14 as a guideline to decide for a duration value that offers an appropriate tradeoff for their specific purpose.

7. Experience and Lessons Learned

Throughout our study, we have had valuable experiences and learned lessons regarding experimentation in operational networks. We have also became aware of possible improvements in our measurement infrastructure and experimentation methodology. In the following, we summarize our notes.

Measurement infrastructure: Observing the performance of operational MBB networks from the perspective of dedicated and controllable clients allows for benchmarking different end-to-end network scenarios (e.g., access technologies, mobility modes, roaming status, location), as well as track the evolution of end-user experience in time. However, in order for performance benchmarks to be fair and comparable, the vantage points should also be comparable. For this reason, we have used MONROE nodes as measurement clients in our experiments.

In the course of our study, a number of issues regarding the testbed infrastructure were also brought to light, which can be provided as feedback to the MONROE Alliance. In terms of mobile node deployment, we have seen the importance of establishing continuous mobility. In this regard, deployment in public transport seems to be preferable over long distance transport. However, a mechanism (e.g., software update to the platform scheduler, which makes use of per-node GPS metadata), which allows the execution of selected experiments only while a measurement node is in mobility, would be a more wholesome solution. In terms of stationary node deployment, we have seen that it is important to have a comparable number of nodes in each geographical unit (e.g., country/city) in order for measurement results to be usable for benchmarking. We have also see that a more even distribution of nodes within a country/city could provide better spatial diversity, as opposed to the natural clustering of nodes around universities or research labs. In terms of roaming, we have seen the difficulty of controlling or even identifying roaming scenarios, on a testbed aiming for measure-

ments "in the wild". Managing roaming scenarios could be facilitated through the use of e-SIM's in the future. Finally, in terms of general maintenance, we have established the importance of node inventory management. In a testbed where nodes can be dynamically moved around, commissioned and decommissioned, it is of utmost importance to reserve resources for maintaining an up-to-date and detailed inventory of node locations which preserves a retrospective history.

Longitudinal measurements: Tracing the performance of networks over time and space is important for many network management practices, such as troubleshooting, performance improvement planning, and resource allocation. Section 5 provides a glimpse of factors which can influence these practices, such as diurnal and weekly patterns.

Over the course of our analysis of the impact of spatio-temporal features on network performance, we identified a strong need to experiment more frequently at randomized points in time (instead of relying on "x times a day at specific hours" type of measurements). Measurements at randomized times during the day provide a fuller picture into the time-of-day effects that were investigated in Section 5.2, which are further enhanced by increased frequency (number of measurements per day on each node). In parallel with this conclusion, we have increased the frequency of MONROE-Nettest base experiments up to 6 times a day on all available nodes. We have also increased the number of measurements conducted on mobile nodes, consequently increasing the number of measurements in verified mobility.

Importance of metadata: Understanding why a given network performs in a certain way requires the identification of causality with respect to influence factors. However, with the complex networking stack of today's communication infrastructures, it is almost impossible to isolate the effects of different parameters on end-to-end network performance individually. This is why, collecting metadata such as location, signal coverage, connection mode, and mobility is crucial. Rich context information is necessary for meaningful statistical analysis, which in turn enables the filtering of false correlation and yields true causality relations. The context-based approach adopted in Section 5 demonstrates some of the potential dimensions of influence.

Speedtest tool design: The MONROE-Nettest tool was designed to run as an experiment over the MONROE platform, with its summary results being collected automatically in a SQL database. There was little emphasis on the detailed results (such as the RTT, data rate, and TCP metrics time series), which later proved to be tremendously useful, as demonstrated in Sections 5 and 6. In the course of this study, we have identified possible improvements in terms of output files (e.g., delivering these in a format which would require less pre-processing, to facilitate efforts such as the one described in Section 6, with more flexibility in terms of aggregation dimensions, and a naming scheme which facilitates preliminary analysis by speedup in parsing), as well as possible additional configuration parameters. We are continuously improving our tool in this regard.

8. Future Outlook

In the following, we present a brief discussion of potential open dataset applications and open challenges that can be addressed in future work.

8.1. Further Dataset Exploitation Potential

Open MBB datasets, such as the one we describe and characterize in Sections 4.2 and 5, can be essential resources for supporting network research. Following up on the use case presented in Section 6, below is a list of potential compelling applications.

Studying external impacts on broadband networks: Tracing the performance of networks over time can also have a wider multi-disciplinary scope and impact. Recent studies such as [55, 56], which provide an analysis of and insights from end-user behaviour and experience during the COVID-19 pandemic, are but examples of such outreach. It is therefore of utmost importance to have established measurement mechanisms in place, which can facilitate the longitudinal collection of network performance information, as well as open datasets such as ours, that can be used to derive technical as well as non-technical insights from periods of interest.

Emulation of MBB networks: Having repeated measurements of network performance from a different number of MNOs over a long period of time provides the possibility to derive detailed models regarding their characteristics. A recent study [57] demonstrates this by developing a mobile network emulator, using MONROE-Nettest measurement profiles to create Kernel Density Estimation models and deploying them on the `tc-netem` tool. Authors manage to emulate dynamic network situations successfully, offering realistic network emulation in which both typical behavior and network variability are accurately recreated. They validate their models with an independent dataset of HTTP download measurements.

Training of ML predictors: Real, so called "in the wild" measurements from operational networks help train ML based predictors for different network performance metrics, such as data rate and latency [7, 46].

Mobile network traces: TCP-based data rate measurements are tremendously useful in generating realistic (mobile) network traces for use in emulation/simulations. A common use case for such traces is the testing of bitrate adaptation algorithms designed for HTTP Adaptive Streaming (HAS) applications. For instance, authors in [58] provide commute path bandwidth traces from 3G networks, which has been widely used in research, but since its publication has become relatively obsolete due to its lack of 4G coverage, as well as high speed mobility scenarios such as measurements along national railways, which we provide in our dataset. Measurements in high mobility have shown to be tremendously useful in identifying network and protocol (e.g., TCP congestion control algorithm) performance optimizations [59].

8.2. Open Challenges

With shifting MBB usage paradigms, increasing practicality of ML based applications, and the approach of next generation

networks, there are many open challenges that can be addressed in future work.

Testbed management: One of the biggest challenges of managing large mobile testbeds is to keep the hardware in the platform up-to-date with developing technologies. However, there is a tradeoff between keeping the testbed infrastructure up-to-date and having comparable longitudinal measurements over time. For instance, updating measurements nodes with 4G capable wireless modems to 5G capable ones would enhance the capabilities of the platform, but none of the previously collected datasets (even if all measurements are in 4G for both) would not be comparable. A second challenge regarding testbeds is the inherent lack of human interaction. Although primarily desirable, this feature blocks interactive measurements in the sphere of QoE testing (e.g., video streaming applications).

Measurements in mobility: One of the biggest challenges for future research is conducting efficient experiments in mobility. The need for shorter test durations, higher accuracy, and analysis mechanisms to handle (cell and/or technology) handovers as well as roaming scenarios, need to be explored further.

ML applications: In Section 6, we demonstrate the efficiency of simple linear regression in exploring and utilizing our open dataset. Such analyses could be extended with different ML algorithms, as well as deeper focus on time series analysis (e.g., with Long Short Term Memory (LSTM) networks). The challenge of managing the TCP slow start period on multiple parallel flows, while aggregating data rate accurately, remains to be open.

Transport protocols: With the increasing prevalence of Google services over the Internet such as YouTube video streaming, as well as dedicated research efforts, QUIC is rapidly gaining popularity over TCP as the transport protocol of choice for many applications. As a protocol operating in user space, QUIC allows for configurations to be more customized, and therefore speed measurements to be made more adaptive, down to individual connections. The possibility of such "user-based adaptation" is an avenue that should be further explored by researchers. Additionally, it is possible to explore the performance implications of using different TCP congestion control algorithms, such as Bottleneck Bandwidth and Round-trip propagation time (BBR), in connection with different mobility scenarios [59], as MONROE-Nettest is able to capture and record all TCP configurations as detailed metadata.

Next generation networks: The upcoming 5G networks will bring forth higher data rates for MBB, causing a more prominent data volume consumption tradeoff in speed testing, as well as a higher number of simultaneously connected devices, possibly changing typical congestion patterns in the Internet. Such developments make intensive research efforts necessary, in order to identify and address possible pitfalls of carrying over today's speed testing paradigms directly into the future.

9. Conclusion

In this paper, we instrument MONROE-Nettest, an open source speedtest tool, for testbed based experimentation in operational mobile networks. We run an extensive longitudinal measurement campaign with this tool over the MONROE testbed, spanning more than 6 MBB network configurations in 2 countries over 2 years, and provide our experiment results together with rich metadata as open data. We characterize the open dataset in detail, as well as derive insights from it regarding the impact of network context, spatio-temporal effects, roaming, and mobility on network performance. We describe our experiences about conducting speedtest measurements in MBB, and discuss challenges associated with large scale testbed experimentation in operational MBB networks. Tackling one of these challenges further, we leverage a ML based algorithm to implement a framework for minimizing data consumption through adaptive speedtest duration. Lastly, we describe the lessons we have learned, as well as provide an overall discussion of how open datasets such as ours can support MBB research, and comment on open challenges, in the hope that these can serve as discussion points for future work.

Acknowledgments

This work was supported by the European Union H2020-ICT research and innovation programme under grant agreement No. 644399 (MONROE), and by the Norwegian Research Council project No. 250679 (MEMBRANE).

References

[1] FCC, Measuring broadband America.
URL `https://www.fcc.gov/general/measuring-broadband-america`

[2] BEREC, Net neutrality tools.
URL `https://net-neutrality.tools`

[3] P. Eardley, A. Morton, M. Bagnulo, P. Aitken, A. Akhter, A framework for large-scale measurement of broadband performance (LMAP).
URL `https://tools.ietf.org/html/rfc7594`

[4] O. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. Safari Khatouni, M. Mellia, M. A. Marsan, Experience: An open platform for experimentation with commercial mobile broadband networks, in: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, MobiCom '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 70–78. doi:10.1145/3117811.3117812.
URL `https://doi.org/10.1145/3117811.3117812`

[5] M. Peón-Quirós, V. Mancuso, V. Comite, A. Lutu, O. Alay, S. Alfredsson, J. Karlsson, A. Brunstrom, M. Mellia, A. Safari Khatouni, T. Hirsch, Results from running an experiment as a service platform for mobile networks, in: Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation amp; CHaracterization, WiNTECH '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 9–16. doi:10.1145/3131473.3131485.
URL `https://doi.org/10.1145/3131473.3131485`

[6] C. Midoglu, L. Wimmer, A. Lutu, Alay, C. Griwodz, Monroe-nettest: A configurable tool for dissecting speed measurements in mobile broadband networks, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2018, pp. 342–347. doi:10.1109/INFCOMW.2018.8406836.

[7] K. Kousias, Alay, A. Argyriou, A. Lutu, M. Riegler, Estimating downlink throughput from end-user measurements in mobile broadband networks, in: 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2019, pp. 1–10. doi:10.1109/WoWMoM.2019.8792968.

[8] Mobile Systems and Analytics Department at SimulaMet, Large scale speedtest experimentation in mobile broadband networks.
URL https://mosaic-simulamet.com/large-scale-speedtest-experimentation-in-mobile-broadband-networks/

[9] A. Nikravesh, D. R. Choffnes, E. Katz-Bassett, Z. M. Mao, M. Welsh, Mobile network performance from user devices: A longitudinal, multidimensional analysis, in: M. Faloutsos, A. Kuzmanovic (Eds.), Passive and Active Measurement, Springer International Publishing, Cham, 2014, pp. 12–22.

[10] A. Elmokasfi, A. Kvalbein, D. Baltrunas, J. Werme, E. Arge, Robusthet i norske mobilnett: Tilstandsrapport 2014, Tech. rep. (2015).
URL https://www.simula.no/sites/default/files/crna2014-5.pdf

[11] A. Elmokashfi, A. Kvalbein, D. Baltrunas, Norske mobilnett i 2015, Tech. rep. (2016).
URL https://www.simula.no/sites/default/files/crna-2015.pdf

[12] A. Elmokashfi, A. Kvalbein, D. Baltrunas, Norske mobilnett i 2016, Tech. rep. (2017).
URL https://www.simula.no/sites/default/files/crna-report2016.pdf

[13] A. Elmokashfi, A. Kvalbein, D. Baltrunas, C. Jarvis, Norske mobilnett i 2017, Tech. rep. (2018).
URL https://www.simula.no/sites/default/files/crna-2017.pdf

[14] A. Elmokashfi, A. Kvalbein, T. Dreibholz, C. Jarvis, Norske mobilnett i 2018, Tech. rep. (2019).
URL https://www.simula.no/sites/default/files/crna-2018_0.pdf

[15] A. Elmokashfi, A. Kvalbein, M. Christiansson, A. S. Al-Selwi, T. Dreibholz, C. Midoglu, Norske mobilnett i 2019, Tech. rep. (2020).
URL https://www.simula.no/sites/default/files/norske_mobilnett_i_2019_1.pdf

[16] A. Karygiannis, E. Antonakakis, mLab: An ad hoc network test bed, in: CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006., Vol. 2, 2006, pp. 1312–1313.

[17] RIPE, RIPE Atlas.
URL https://atlas.ripe.net

[18] M. Isah, A. Phokeer, J. Chavula, A. Elmokashfi, A. S. Asrese, State of internet measurement in Africa - a survey, in: R. Zitouni, M. Agueh, P. Houngue, H. Soude (Eds.), e-Infrastructure and e-Services for Developing Countries, Springer International Publishing, Cham, 2020, pp. 121–139.

[19] A. M. Mandalari, A. Lutu, A. Custura, A. Safari Khatouni, O. Alay, M. Bagnulo, V. Bajpai, A. Brunstrom, J. Ott, M. Mellia, G. Fairhurst, Experience: Implications of roaming in europe, in: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, MobiCom '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 179–189. doi:10.1145/3241539.3241577.
URL https://doi.org/10.1145/3241539.3241577

[20] C. Jarvis, C. Midoglu, A. Lutu, O. Alay, Visualizing mobile coverage from repetitive measurements on defined trajectories, in: 2018 Network Traffic Measurement and Analysis Conference (TMA), 2018, pp. 1–6. doi:10.23919/TMA.2018.8506487.

[21] S. Bauer, D. Clark, W. Lehr, Understanding broadband speed measurements, TPRC (2010).

[22] Ookla, Speedtest by Ookla - the global broadband speedtest.
URL https://www.speedtest.net

[23] OpenSignal, OpenSignal: Mobile analytics and insights.
URL https://www.opensignal.com

[24] Zafaco, Summary - kyago.
URL https://kyago.com/kyago/breitbandtest-web-app/zusammenfassung/

[25] RTR, RTR - NetTest.
URL https://www.netztest.at

[26] NKOM, Nettfart.no - test kapasiteten paa nettoppkoblingen din.
URL https://www.nettfart.no

[27] HAKOM, HAKOMetar.
URL https://www.hakom.hr/default.aspx?id=1144

[28] CZ.NIC, NetMetr.
URL https://www.netmetr.cz/en/

[29] AKOS, AKOS Test Net.
URL https://www.akostest.net/en/

[30] RATEL, RATEL Nettest.
URL https://nettest.ratel.rs/en/

[31] MLAB, MobiPerf.
URL https://www.measurementlab.net/tests/mobiperf/

[32] Netradar, Netradar.
URL https://www.netradar.com

[33] C. Midoglu, P. Svoboda, Opportunities and challenges of using crowdsourced measurements for mobile network benchmarking a case study on RTR open data, in: 2016 SAI Computing Conference (SAI), 2016, pp. 996–1005. doi:10.1109/SAI.2016.7556101.

[34] K. Kousias, C. Midoglu, O. Alay, A. Lutu, A. Argyriou, M. Riegler, The same, only different: Contrasting mobile operator behavior from crowdsourced dataset, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–6. doi:10.1109/PIMRC.2017.8292203.

[35] A. Schwind, C. Midoglu, Alay, C. Griwodz, F. Wamser, Dissecting the performance of youtube video streaming in mobile networks, International Journal of Network Management 30 (3) (2020) e2058, e2058 nem.2058. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.2058, doi:10.1002/nem.2058.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2058

[36] Y. Liu, J. Y. B. Lee, An empirical study of throughput prediction in mobile data networks, in: 2015 IEEE Global Communications Conference (GLOBECOM), 2015, pp. 1–6. doi:10.1109/GLOCOM.2015.7417858.

[37] M. Mirza, J. Sommers, P. Barford, X. Zhu, A machine learning approach to TCP throughput prediction, in: Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '07, Association for Computing Machinery, New York, NY, USA, 2007, p. 97–108. doi:10.1145/1254882.1254894.
URL https://doi.org/10.1145/1254882.1254894

[38] A. Samba, Y. Busnel, A. Blanc, P. Dooze, G. Simon, Instantaneous throughput prediction in cellular networks: Which information is needed?, in: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017, pp. 624–627. doi:10.23919/INM.2017.7987345.

[39] B. Wei, W. Kawakami, K. Kanai, J. Katto, S. Wang, TRUST: A TCP throughput prediction method in mobile networks, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6. doi:10.1109/GLOCOM.2018.8647390.

[40] J. Riihijarvi, P. Mahonen, Machine learning for performance prediction in mobile cellular networks, IEEE Computational Intelligence Magazine 13 (1) (2018) 51–60. doi:10.1109/MCI.2017.2773824.

[41] T. Linder, P. Persson, A. Forsberg, J. Danielsson, N. Carlsson, On using crowd-sourced network measurements for performance prediction, in: 2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 2016, pp. 1–8.

[42] C. Rattaro, P. Belzarena, Throughput prediction in wireless networks using statistical learning, in: LAWDN - Latin-American Workshop on Dynamic Networks, INTECIN - Facultad de Ingeniería (U.B.A.) - I.T.B.A., Buenos Aires, Argentina, 2010, p. 4 p.
URL https://hal.inria.fr/inria-00531743

[43] V. Raida, P. Svoboda, M. Rupp, Constant rate ultra short probing (CRUSP) measurements in LTE networks, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018, pp. 1–5. doi:10.1109/VTCFall.2018.8690838.

[44] V. Raida, P. Svoboda, M. Kruschke, M. Rupp, Constant rate ultra short probing (CRUSP): Measurements in live LTE networks, in: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6. doi:10.1109/ICC.2019.8761179.

[45] C. Maier, P. Dorfinger, J. L. Du, S. Gschweitl, J. Lusak, Reducing con-

sumed data volume in bandwidth measurements via a machine learning approach, in: 2019 Network Traffic Measurement and Analysis Conference (TMA), 2019, pp. 215–220. doi:10.23919/TMA.2019.8784575.

[46] A. Safari Khatouni, F. Soro, D. Giordano, A machine learning application for latency prediction in operational 4g networks, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, pp. 71–74.

[47] K. Gao, J. Zhang, Y. Richard Yang, J. Bi, Prophet: Fast accurate model-based throughput prediction for reactive flow in DC networks, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 2018, pp. 720–728. doi:10.1109/INFOCOM.2018.8486372.

[48] N. Bui, F. Michelinakis, J. Widmer, A model for throughput prediction for mobile users, in: European Wireless 2014; 20th European Wireless Conference, 2014, pp. 1–6.

[49] V. Mancuso, M. P. Quirós, C. Midoglu, M. Moulay, V. Comite, A. Lutu, Ö. Alay, S. Alfredsson, M. Rajiullah, A. Brunström, et al., Results from running an experiment as a service platform for mobile broadband networks in Europe, Computer Communications 133 (2019) 89–101.

[50] C. Midoglu, L. Wimmer, P. Svoboda, Server link load modeling and request scheduling for crowdsourcing-based benchmarking systems, in: 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), 2016, pp. 988–994. doi:10.1109/IWCMC.2016.7577193.

[51] MONROE, MONROE-Nettest docker container.
URL docker.monroe-system.eu/monroe/monroe-nettest/image

[52] Cisco, Cisco annual internet report (2018–2023) white paper, Tech. rep. (2020).
URL https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[53] M. Kargar Bideh, A. Petlund, C. Griwodz, I. Ahmed, R. behjati, A. Brunstrom, S. Alfredsson, Tada: An active measurement tool for automatic detection of aqm, in: Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS'15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 2016, p. 55–60. doi:10.4108/eai.14-12-2015.2262684.
URL https://doi.org/10.4108/eai.14-12-2015.2262684

[54] G. A. F. Seber, A. J. Lee, Linear Regression Analysis, Vol. 329 of Wiley Series in Probability and Statistics, John Wiley & Sons, 2012.

[55] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, G. Smaragdakis, The lockdown effect: Implications of the COVID-19 pandemic on internet traffic, in: Proceedings of the ACM Internet Measurement Conference, IMC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1–18. doi:10.1145/3419394.3423658.
URL https://doi.org/10.1145/3419394.3423658

[56] M. Rajiullah, A. Safari Khatouni, C. Midoglu, O. Alay, A. Brunstrom, C. Griwodz, Mobile network performance during the COVID-19 outbreak from a testbed perspective, in: Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation amp; Characterization, WiNTECH'20, Association for Computing Machinery, New York, NY, USA, 2020, p. 110–117. doi:10.1145/3411276.3412194.
URL https://doi.org/10.1145/3411276.3412194

[57] M. Trevisan, A. Safari Khatouni, D. Giordano, ERRANT: Realistic emulation of radio access networks, Computer Networks 176 (2020) 107289. doi:https://doi.org/10.1016/j.comnet.2020.107289.
URL http://www.sciencedirect.com/science/article/pii/S1389128620301420

[58] H. Riiser, P. Vigmostad, C. Griwodz, P. Halvorsen, Commute path bandwidth traces from 3G networks: Analysis and applications, in: Proceedings of the 4th ACM Multimedia Systems Conference, MMSys '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 114–118. doi:10.1145/2483977.2483991.
URL https://doi.org/10.1145/2483977.2483991

[59] J. Wang, Y. Zheng, Y. Ni, C. Xu, F. Qian, W. Li, W. Jiang, Y. Cheng, Z. Cheng, Y. Li, X. Xie, Y. Sun, Z. Wang, An active-passive measurement study of TCP performance over LTE on high-speed rails, in: The 25th Annual International Conference on Mobile Computing and Networking, MobiCom '19, Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3300061.3300123.
URL https://doi.org/10.1145/3300061.3300123