Results from Running an Experiment as a Service Platform for Mobile Broadband Networks in Europe☆

Vincenzo Mancuso[a,*], Miguel Peón Quirós[a,b], Cise Midoglu[c], Mohamed Moulay[a,d], Vincenzo Comite[a,e], Andra Lutu[c], Özgü Alay[c], Stefan Alfredsson[f], Mohammad Rajiullah[f], Anna Brunström[f], Marco Mellia[g], Ali Safari Khatouni[g], Thomas Hirsch[h]

[a]*IMDEA Networks Institute, Madrid, Spain*
[b]*EPFL, Lausanne, Switzerland*
[c]*Simula Research Laboratory, Oslo, Norway*
[d]*University Carlos III of Madrid, Spain*
[e]*Università La Sapienza, Rome, Italy*
[f]*Karlstad University, Karlstad, Sweden*
[g]*Politecnico di Torino, Turin, Italy*
[h]*Celerway, Oslo, Norway*

Abstract

In this article we present a selection from a vast range of experiments run with MONROE, our open *experiment as a service* (EaaS) platform for measurements and experimentation in Mobile Broadband Networks. We show that the platform can be used to benchmark network performance in a repeatable and controlled manner thanks to the collection of a rich set of geotagged metadata and the execution of discretionary user experiments. Indeed, with the sheer amount of data collected from 12 commercial mobile operators across Europe, MONROE offers an unprecedented opportunity to monitor, analyze and ultimately

☆Work funded by the EU H2020 research and innovation programme under grant agreement No. 644399 (MONROE). The work of V. Mancuso was supported by the Ramon y Cajal grant (ref: RYC-2014-16285) from the Spanish Ministry of Economy and Competitiveness.
A preliminary version of this article has appeared in the proceedings of the ACM WiNTECH 2017 Workshop [1].

*Corresponding author
Email addresses: vincenzo.mancuso@imdea.org (Vincenzo Mancuso), miguel.peon@epfl.ch (Miguel Peón Quirós), cise@simula.no (Cise Midoglu), mohamed.moulay@imdea.org (Mohamed Moulay), vincenzo.comite@gmail.com (Vincenzo Comite), andra@simula.no (Andra Lutu), ozgu@simula.no (Özgü Alay), stefan.alfredsson@kau.se (Stefan Alfredsson), mohammad.rajiullah@kau.se (Mohammad Rajiullah), anna.brunstrom@kau.se (Anna Brunström), mellia@polito.it (Marco Mellia), ali.safari@polito.it (Ali Safari Khatouni), thomas.hirsch@celerway.com (Thomas Hirsch)

improve the status of current and future mobile broadband networks. Besides, we show how flexibly the platform allows combining metadata and experimental data series during the experiments or by means of post-processing, and show results produced by our own experiments as well as comment on results obtained by external research groups and developers that have been granted access to our platform.

Keywords: Mobile broadband; EaaS, Measurements; Network experimentation; Large testbed; Metadata; Performance analysis; Repeatability and reproducibility.

---

1. Introduction

The field of networking offers the possibility of gathering large volumes of information from network elements and end hosts. Analyzing these data is crucial to understand how networks perform under different usage patterns and adapt them to future requirements. This is particularly important for mobile broadband networks (MBBs), which are the segment with the strongest growth forecast and higher variability in operating conditions. Two main challenges arise when trying to analyze the performance and reliability of MBBs: The difficulty of obtaining systematic data from reliable repetition of experiments on commercial operational MBB networks, and sifting through the big amount of variables that can be monitored and measured.

MONROE[1] is a Europe-wide experiment oriented network counting on more than 200 custom measurement devices (or *nodes*), designed to enable collection and analysis of the characteristics of commercial mobile broadband networks and execution of discretionary experiments from external researchers. The platform nodes operate under a wide variety of conditions, the nodes being deployed aboard trains, buses and delivery trucks, or inside residential homes and labo-

[1]MONROE is a FIRE+ project funded by the European Union's H2020 research and innovation programme. For more information, please visit https://www.monroe-project.eu/

ratories. Nodes are co-located in pairs, where one node connects to two mobile providers using customer-grade commercial subscriptions, and the other connects to a third operator and potentially to a WiFi network. Both nodes can connect to Ethernet where available.

The testbed performs periodic passive and active measurements and continuously monitors the status of the MBB networks through metadata collection. The collected metadata are centrally stored in a NoSQL database to ensure scalability past billions of records. We offer to the community the unique possibility of accessing our curated dataset through periodic data dumps, which enable data analysis across all the nodes and lifespan of the platform. Additionally, we encourage external experimenters to devise novel experiments and add to the diversity of MONROE open data.

The following is a list of the main characteristics and innovations of MONROE, which exposes software services and physical nodes to plan and perform MBB measurements, hence it is an *Experiment as a Service* (EaaS) platform.

Large-scale deployment in diversified scenarios: MONROE nodes are being deployed across Norway, Sweden, Italy and Spain, with external partners currently deploying additional nodes in Germany, Greece, France, Portugal, Slovenia and the UK. Some nodes have stationary locations in dense urban areas, while a significant number (more than 110 at the time of writing) operate aboard public inter-city trains, buses and delivery trucks. Whereas trains traverse large distances, sometimes at high speeds, buses cover urban areas. Both settings enable us to collect a unique dataset under mobility scenarios along the fix routes of those vehicles. Nodes aboard delivery trucks, which traverse both urban and rural areas without fixed routes, complement the previous dataset.

Open experimentation platform on commercial cellular operators: MONROE is an open platform that allows authenticated researchers to run their own custom experiments on commercial MBB networks. Researchers can then opt to add their data to the MONROE open dataset, increasing its diversity and allowing us to look past performance metrics and metadata. Notable examples are a Web performance experiment and video QoE measurements [2], which are being evaluated for inclusion in the set of periodic measurements run on the nodes. In addition to the actual data, experiment source code and supporting material for those wanting to create new experiments on MONROE are also openly available.[2]

Consistency and repeatability: MONROE provides a uniform hardware and software environment to measure and monitor MBB networks at fixed locations and times. Furthermore, the public transportation vehicles that host MONROE nodes ensure fairly repeatable routes for mobility experiments. Even more, they repeat the same itineraries several times a day at different hours (i.e., mixing peak and normal hours) and on different days (i.e., weekdays and weekends). This provides the dataset with a rich spatio-temporal dimension, which is key to enable the comparison of different measurements over different operators, places and times of day.

Metadata-rich dataset: Each MONROE node is instrumented to periodically measure the performance of its MBB providers. They continuously gather metadata, including, for example, location, signal strength and link technology for each network provider. Additionally, several basic speed and network probing tests are executed periodically to asses network performance. Since MONROE does not involve real users (which usually entail privacy protection restrictions), rich metadata collection, including geo-temporal tagging, is possible, which enables the evaluation of mobile services under mobility. In particular, MONROE collection of data enables purely off-line experiments for analysis of MBB network performance.

In the rest of this paper, we start by describing the design of the MONROE EaaS platform in Section 2. In Section 3, we first present the tools offered to experimenters and currently available *template* experiments, then we showcase the possibilities that MONROE opens by presenting a selection of experiments

[2]All stable pieces of open source code produced in MONROE are available on `github` at `https://github.com/MONROE-PROJECT/Experiments`, whereas a complete user manual is made available at `https://github.com/MONROE-PROJECT/UserManual`

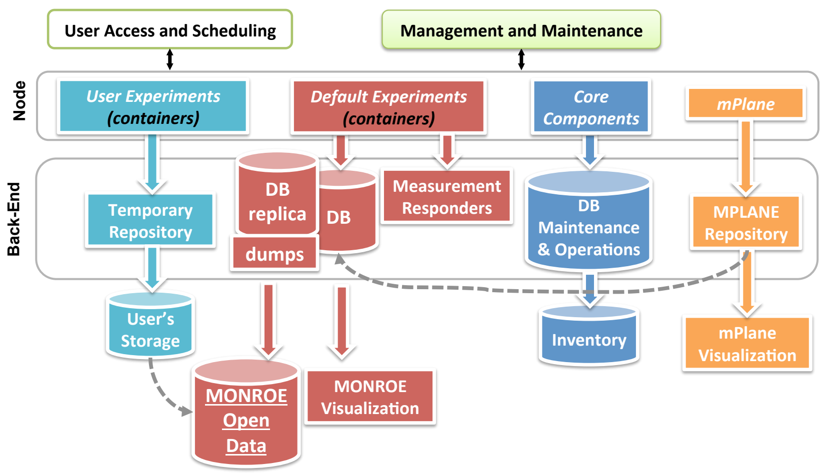


Figure 1: MONROE system design. Researchers access the system through the Web user interface and scheduler, or directly through the various repositories and data bases. Several passive (metadata, mPlane, etc.) and active (RTT, HTTP bandwidth, etc.) probes monitor continuously network usage and performance.

run by us and by several external groups. Those examples aim to entice other researchers to exploit the data gathered by our platform in innovative ways or to design their own experiments and so contribute to improve our overall knowledge on the behavior of MBB networks. We comment on related experimental work and platforms in Section 4 and conclude the article in Section 5.

2. EaaS platform design and implementation

The MONROE EaaS platform was designed with the purpose of collecting, storing and offering open access to large amounts of diverse mobile network data, and providing an EaaS platform for the execution of discretionary experiments by external researchers. Therefore, enriching measurement data with abundant context information (metadata), and enabling a wide variety of experiments, are the two key aspects that have steered the platform design since its inception. Figure 1 offers a high-level overview of the complete MONROE platform design.

An introduction to the platform was presented in [3], while its detailed description and the experience of operating it are presented in [4]. Therefore, here we briefly present the platform components and focus on the processes of collection and storage of measurement results and the concrete implementation choices made during the platform design. The system design includes four main groups of components distributed across nodes and backend, as shown by the color code adopted in Figure 1.

The "red" component is responsible for MONROE default experiments, each using an isolated Linux Docker container [5]. Default MONROE experiments include, for example, periodic ping measurements for connectivity survey, HTTP downloads from a series of targets under our control, or Web performance measurements. The results of these default experiments and the collected metadata are transferred as JSON files to the main MONROE server via `rsync` over SSH channels. Once at the server, the JSON files are stored in a NoSQL database. Off-line data analysis can happen both at the server side in the form of database queries or at the experimenter's side (with custom applications) if further processing is required. Since data are the main asset of the platform, we implement several backup and duplication mechanisms to provide data safety and access redundancy. A visualization solution facilitates the surveillance of the platform health and its available resources in near real-time.

Beside default experiments, MONROE allows authenticated external researchers to access the platform via the Web user interface and deploy their own custom experiments. This is the "azure" component of Figure 1. Separate storage for the results of user experiments is offered in a temporary repository accessible through the platform Web user interface. We encourage users to make their results public and include them in the MONROE open dataset.

In addition to default and external experiments, each node runs Tstat [6], a passive traffic analysis tool connected to the mPlane measurement platform [7]. Tstat generates a series of logs that the nodes send to the mPlane repository, from where users can consume the data using the mPlane visualization solution. This is the "orange" component in Figure 1. Note that Tstat data is also imported to the MONROE DB, as shown in the figure.

A fourth component, the "bue" one in Figure 1, has been designed for dealing with node connectivity and software management of the platform.

As shown in the upper part of Figure 1 access to the platform is guaranteed to experimenters by means of a user access portal, and experiments are automatically loaded by a global scheduler that enforces and activates the Docker containers provided by the experimenters and carrying the experimental code. Thus, the entire architecture is transparent to the end-users, i.e., the experimenters. Moreover, platform maintainers have direct and exclusive access to the nodes and to the MONROE back-end.

2.1. Node instrumentation

MONROE nodes collect four types of information:

1. Metadata: This includes network parameters (RSSI, cell identifiers, link technology, etc.), node location and speed (GPS), node working parameters (CPU temperature, processing load, etc.) and node events (watchdogs).

2. Connectivity and latency measurements: Basic active measurements are run in a container that collects statistics on ICMP packets sent towards fixed destinations (UDP/TCP RTT will be added as future extensions).

3. MONROE and user experiments: Experimenters define Docker containers to run their measurements in isolation. Some containers are scheduled periodically to estimate available bandwidth estimation, to track routes to and from specific targets in the network, etc. Other containers are scheduled upon the request of the experimenters.

4. Passive traffic monitoring: TCP flows are captured and analyzed by means of the Tstat measurement suit. MONROE nodes include a Tstat probe in a dedicated container, in which all MBB interfaces are monitored and where per-flow statistics are computed and subsequently published in the mPlane and MONROE databases.

The differentiation between the aforementioned types of data responds to their distinct natures and purposes. In that way, passive metadata can be gathered at the nodes with minimal impact on any experiments; thus, they are recorded on a continuous basis. Similarly, the passive mPlane Tstat probe,

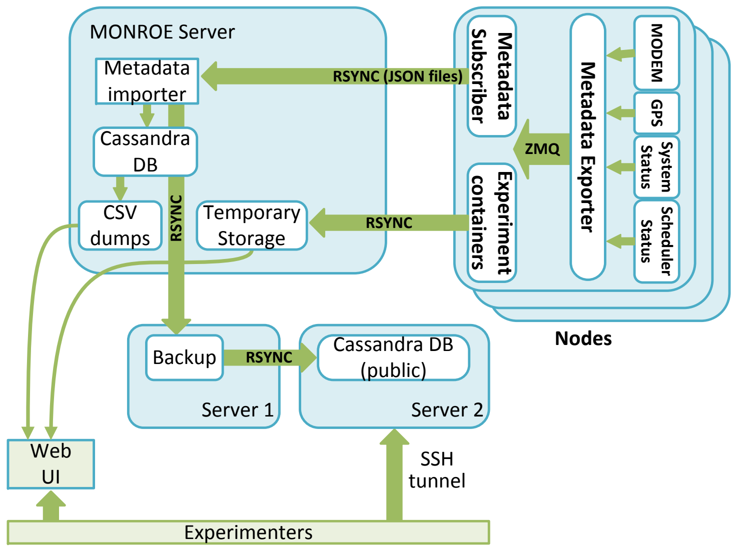


Figure 2: Flow of information in the MONROE platform.

which produces low processing load, runs continuously. Background experiments such as end-to-end delay or round-trip time (RTT) measurements may create a moderate (controlled) interference with other experiments; however, the value obtained by gathering these data are worth their cost. Experimenters are made aware of those background experiments; furthermore, we are evaluating a mechanism to allow them to pause their execution. Finally, some MONROE experiments such as bandwidth measurements might produce a higher impact on user experiments. Therefore, those experiments are not scheduled concurrently with any user's ones. Indeed, each user experiment runs in exclusivity with respect to experiments from any other users.

2.2. Data flows

Figure 2 shows the different flows of information through the platform, since it is generated in a node until it is collected and stored in our databases for later analysis. MONROE nodes implement a metadata distribution mechanism based on a publish/subscribe model. Experiments running in the nodes can subscribe to different information "topics" to monitor system status and events such as network interface (dis)connections, link technology changes or GPS location variations. This flexible design eases the implementation of each platform component as data producers do not need to keep track of their clients, and new data consumers can choose the information topics they are interested on without caring about the details of the producers.

Independently of their origin, all data items are transferred to the MONROE servers via `rsync` over SSH. Once at the server, each item is processed and stored according to its nature: Metadata, the results of the MONROE experiments and Tstat measurements, which arrived as JSON files, are stored in a NoSQL database, whereas the results of user experiments are temporarily kept at a repository for easy access through a Web user interface.

In the server, several scripts create backups of the database contents and a dump of the database in CSV format is produced daily; experimenters may use those 24 h feeds if their experiments are focused on small periods of time. Furthermore, a secondary copy of the database is updated every day for direct access by external researchers. That secondary copy is not a normal database "replica" to avoid the risk that accidental (or malicious) modifications to the (open) database spread to the primary one. The daily CSV dumps are available for direct download to registered users through the Web user interface; access to the (secondary) database is provided to external researchers via SSH tunnels.

2.3. At the node side

At the node side, metadata distribution is implemented in a publish/subscribe pattern using ZMQ.[3] The metadata stream is available for experiments during their execution using the ZMQ subscription mechanisms. Metadata entries are generated in a single-line JSON format, which eases human analysis. Every data entry is labeled with a "topic" field; consumers may subscribe to the whole stream of metadata or just to some topics. The metadata subscriber module runs in the nodes and subscribes to all the topics, writing

[3] http://zeromq.org/

JSON entries to files in a special file system location. A synchronization process transfers those files to the MONROE server when no other active, periodic, or user-defined experiment is running.

Regarding node stability, several monitoring and recovery methods ensure that they remain online and capable of executing experiments. Node stability is ensured via lightweight virtualization (by means of Docker containers), thus guaranteeing a clean environment for each experiment. Several surveillance mechanisms (watchdogs) in the nodes can force a complete reinstallation of the operating system and environments if they detect system malfunctions such as filesystem corruption.

2.4. At the server side

Information received from the nodes in JSON format is stored at the server in a NoSQL database. The choice of a NoSQL solution was based on the need to permanently store a potentially very large dataset consisting of billions of entries. As a quick calculation to illustrate the scale of the dataset, RTT measurements are executed for each of the three MBB interfaces of each node every second. Therefore, $3 \times 3\,600\,\mathrm{s} \times 24\,\mathrm{h} \times 365\,\mathrm{days} \times 150\,\mathrm{nodes} = 14\,191 \times 10^6$ entries are stored in the database every year, only for RTT measurements. Based on the concrete storage and access needs of MONROE, Apache Cassandra[4] was chosen as the system NoSQL database for its scaling abilities, both in performance and storage capacity. If the space available in a machine is exhausted, new space may be added simply by configuring a new replica. Additionally, Cassandra is a mature technology that offers access drivers for multiple programming languages and production-grade tools for data analytics, widening access options for researchers. Besides, several Python scripts produce a backup of the JSON files received at the server and a daily CSV dump of the database. Those results are transferred to a backup server that provides off-site backups. The copy of the database accessible to external researchers is hosted in an independent

[4]http://cassandra.apache.org/

server, thus avoiding performance interferences with the main database.

2.5. Access to data

The metadata produced by the nodes can be accessed in several ways. First, experiments may access the metadata stream during execution using the ZMQ subscription mechanisms. In this way, they can monitor and react to events such as interface reconnections or link technology and signal strength changes for each MBB at run-time. Second, researchers may access the database (or the CSV dumps) to correlate their results with the metadata matching by the corresponding timestamps. As an example, the results of an experiment may be related to the network conditions during its execution, even if at that time not all the metadata was checked online. Researchers may also import the CSV dumps into their own tools for more specific data analyses.

2.6. User access and experiment scheduling

MONROE enables user access to the experimental platform through a user-friendly interface built on an AngularJS-based Web portal. The platform is open, although authentication is required. In particular, as part of the MONROE federation with the Fed4FIRE initiative of the European Commission,[5] MONROE user access follows Fed4FIRE specifications in terms of authentication and provisioning of resources. Hence, the MONROE portal allows to access the MONROE scheduler, which is a server in charge of setting up the experiments without requiring the users to directly interact with the nodes (i.e., no SSH access to the node environment). The scheduler ensures that there are no conflicts between users when running their experiments and assigns resources to each user.

The scheduling system consists of two parts. A scheduling server runs on a MONROE server behind an Nginx proxy and uses an SQLite 3 database to store user roles, node and experiment status, and schedules. In addition to

[5] `http://www.fed4fire.eu/`

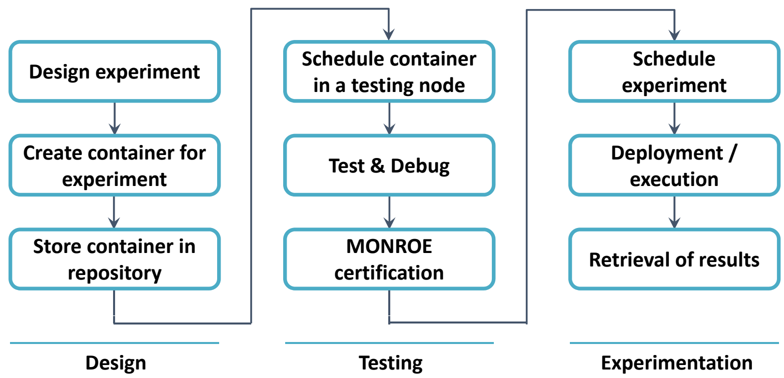


Figure 3: Experiment workflow covering the design, test and experimentation phases.

Fed4FIRE-compatible APIs, it offers a REST API that can be accessed through the Web user interface or directly through the Nginx proxy if users develop their own access scripts. The client part of the scheduler runs on MONROE nodes. It periodically contacts the scheduler in the server to send "heartbeats" and traffic statistics, and check for new schedules for the node. When new schedules are available, the scheduler preloads up to three containers, depending on criteria such as available storage in the node and time until schedule. It also schedules the start and stop times of each container using operating system functions. When the time to execute a new container arrives, the operating system executes the container using the Docker tools. Finally, the scheduler monitors the experiments to check if they exceed the allocated resources and to transfer any result files and inform of result codes.

2.7. Experimentation workflow

Figure 3 shows the general workflow of the experiments executed on MONROE nodes. The first step is to design the experiment selecting the appropriate tools. The required files have to be collected in a Docker container, which is submitted to a repository. MONROE offers a set of dedicated testing nodes that can execute containers from any public repository. Once the experiment is ready, it undergoes a certification process in which MONROE administrators check that it is generally safe for execution and move the container to a private repository.

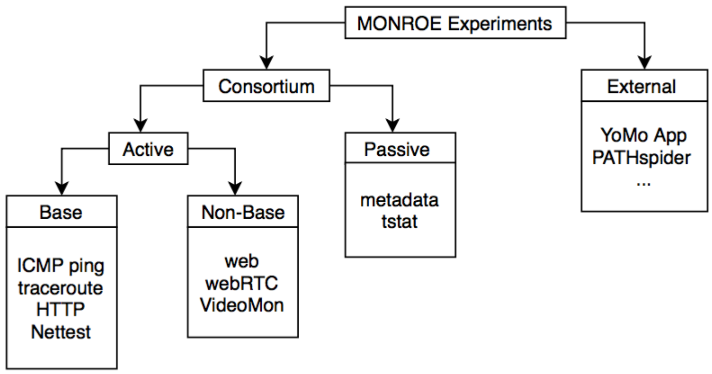


Figure 4: Experiments currently available as services that can be run on the MONROE platform, within an EaaS framework.

Deployed nodes (i.e., real experimentation nodes) can download containers only from the MONROE private repository. Container execution can be scheduled as many times and on as many nodes as required, always subject to quota availability. Using the platform Web interface, users can monitor the progress of all their experiments, including repetitions on multiple nodes. Finally, the results can be downloaded directly from the platform Web page.

3. Experiments

In this section, we list and describe the EaaS templates currently available in form of docker containers and ready for experimentation with the MONROE platform, as well as a number of their use cases for MBB evaluation.

3.1. Experiments currently available as services

There are many experiments available as services on the MONROE platform within an EaaS framework. Figure 4 lists these experiments with respect to their origin and characteristics.

Consortium experiments are provided by the MONROE Consortium and are all available on `github`, as mentioned in the introduction, jointly with detailed instructions on how to configure and run the experiments. For each experiment

Table 1: MONROE metadata topics

Class	**Type**	***Examples***
Node	Sensor	*CPU temperature*
Node	Probe	*Load, memory usage*
Node	Event	*Power up, reboot*
Device	GPS	*GPS coordinates*
Device	Modem	*RSSI, link technology, cell ID, IP address*
Experiment	RTT	*Ping RTT*
Experiment	Bandwidth	*HTTP download throughput*

and the Docker container implementing it, a template is prepared and provided for running the experiment from the MONROE Web interface with a single click, without the need to set any configuration parameters. The default parameters can be modified at will, as documented in the MONROE user manual.

The experiments can be passive or active. Passive experiments collect information in the background without generating additional traffic, whereas active experiments perform new measurements by generating data traffic. Passive experiments run continuously (periodically). Active experiments can either be run periodically with a lower frequency than the passive experiments (those are referred to as "base" experiments), or be available for running at will without a regular schedule ("non-base experiments").

In addition to the consortium experiments, the platform is currently used for research and experimentation by 27 external groups from academy and industry, who run both passive and active experiments.

In what follows, we describe the design and implementation of the most prominent experiments produced by the MONROE consortium and provided as EaaS, and give two examples of experiments designed by external experimenters.

3.1.1. Metadata collection

MONROE nodes passively and continuously generates metadata. Table 1 illustrates the metadata "topics", which are streamed to subscriber entities within the node using ZMQ, as previously explained in Sections 2.2 and 2.3. Metadata are collected and stored in a database for post processing. However, experiments running containers can also have their containers subscribed to any of the metadata topics and use them during their experiments or store them jointly with their results, to easy the joint post processing of data and metadata pertinent to a given experiment. Upon a variation in a monitored value, a new message is sent to subscribers only, so the metadata generation uses limited resources.

3.1.2. TCP flow analysis

One of the Docker containers always present in a MONROE node runs TCP flow analysis in near-real time using Tstat. Tstat is a powerful passive monitoring tool that rebuilds TCP flows reporting more than 100 flow descriptors (e.g., client and server IP and port, RTT, number of retransmissions) and more than a thousand packet level metrics [6]. Therefore the container implements a passive traffic probe that provides insights on the traffic patterns at both the network and the transport levels, offering additional information on the traffic each interface exchanged during an experiment. This container runs continuously and does not interfere with other experiments. Moreover, experimenters can use Graphite[6] to easily navigate through offline logs and store a dashboard showing relevant data within an adjustable time window.

3.1.3. End-to-end delay statistics

This is a base experiment running active and lightweight measurements. It consists in a simple container that pings continuously a few remote targets and records ICMP ping statistics. A variant of this container is also available, in which UDP is used instead of ICMP. Despite being very simple, this experi-

[6] http://graphite.readthedocs.org/en/latest/index.html

ment gives fundamental information about the status of the network and its congestion level.

3.1.4. Route monitoring

MONROE incorporates active traceroute measurements in the set of base experiments, to study routing and to identify middleboxes. The MONROE traceroute experiment aims to compare routing from nodes in different countries and, inside a country, different operators (some of our measurements are performed with SIM cards in roaming that show home-routing patterns). By using Paris Traceroute[7] rather than a simple and legacy traceroute application, these experiments also allow identifying middleboxes and their differences between operators and countries.

3.1.5. Webpage download

To assess basic Web performance figures, one of the experiments available as service in MONROE, running on-demand, is an active experiment using an headless browser that can be run in a node with no need of a monitor. It measures Web performance against multiple popular targets, enabling, for example, the tracking of page load time and its correlation with metadata information.

3.1.6. HTTP download

This is another active base experiment that is periodically scheduled in all nodes. The container tests HTTP download rates using the various available versions of HTTP, and generate statistics about large file downloads. Being data-consuming, this test is not aggressively scheduled, although it is needed to complement the statistics on delay/RTT studied by means of tiny ping packets and on short-lived flows collected with the webpage download experiments described above.

[7] `https://paris-traceroute.net/`

3.1.7. Network speed tests

MONROE-Nettest is a configurable tool for data rate and latency measurements, intended for the study of speed in MBB networks, using active experiments. We choose RMBT by Netztest[8] as the codebase for our client implementation since this is a tool used by most network regulatory authorities in Europe for their crowdsourced measurement applications. Adopting a user experience oriented approach for measuring data rate, these solutions use TCP-based testing with multiple parallel flows. Configurable parameters of the client include the number of flows for downlink and uplink, measurement durations, and measurement server. For the server side, we make sure to keep compatibility with the RMBT, and use the server code from the open-source Open-RMBT project,[9] with only minor changes. We have deployed a network of MONROE-Nettest servers in Europe, including Germany, Norway, Spain, and Sweden for large scale experimentation. MONROE-Nettest[10] is run as a base experiment in the MONROE platform, so it is run periodically on every node and every connected MBB network.

3.1.8. WebRTC streaming

WebRTC is based on web technologies like HTML and JavaScript, and consists on integrating video, audio and data streams belonging to a session using the RTC protocol into a webpage, with no need of plugins and calls to external software. In MONROE, we have developed a Docker container that implements a WebRTC streamer and an IP tunnel handler that makes available a multimedia file over HTTPS. The container then includes a light implementation of WebRTC for EaaS.[11] When the WebRTC container is scheduled and runs on a set of machines, each of them makes a link available for connecting and watching the multimedia file using a Chrome browser acting as WebRTC client. The

[8] `https://www.netztest.at/doc/`

[9] https://github.com/ alladin-IT/open-rmbt commit dfc008de71e321c863716b0d34208159b140c653

[10] `https://github.com/MONROE-PROJECT/Experiments/tree/master/experiments/nettest`

[11] The WebRTC container is available at `docker.monroe-system.eu/deployed/monrtc`

WebRTC container implements active experiments which are not part of the base experiments set.

3.1.9. Adaptive streaming over HTTP

MONROE uses a variant of AStream, which is an open source software written in Python to implement 3 different rate adaptation algorithms for evaluating adaptive streaming over HTTP using DASH.[12] We have adapted the existing AStream framework to the MONROE platform with slight modifications, providing a suitable Docker container which integrates a wrapper. Therefore, this is an active type of experiments, which currently run as a non-base MONROE container. However, this experiment will soon be run as a base experiment within the VideoMon container,[13] which is a combination of the consortium experiment AStream with the external user experiment YoMoApp (more info in Section 3.1.10).

3.1.10. Video QoE with YoMoApp

YoMoApp [2] is an application for YouTube performance monitoring, which allows analyzing mobile network performance with respect to YouTube traffic. It also serves developing optimization solutions and QoE models for mobile HTTP adaptive streaming. The application has been developed by external MONROE experimenters to extend MONROE into the domain of QoE with the design and implementation of a measurement tool for YouTube video streaming sessions. YoMoApp gathers statistics on initial delay, video adaptation over HTTP, HTTP request and response information, and stalling occurrences [8].

3.1.11. Path transparency

This is another example of MONROE container and experiment developed and provided by external experimenters. The container uses *PATHspider* [9] to detect the presence of middleboxes over point-to-point paths. In addition, it

[12]https://github.com/pari685/AStream
[13]https://hub.docker.com/u/videomon/

tests the feasibility of deploying new protocols in the Internet while quantifying the impact of path impairments.

3.2. Selected measurement studies

Next, we present some of the most interesting studies that have been conducted on MONROE using the previously described experiments and/or the MONROE dataset, which, at the time of writing, contained more than 975 M metadata entries, 1882 M RTT and 107 K bandwidth measurements, 38 M Tstat entries and more than 18 K experimenter results. **[To be updated]**

Studies on the MONROE platform can be passive or active. *Passive studies* analyze and use the curated MONROE dataset, which contains metadata, the results of the default experiments and the results of experiments shared by their owners with the broader community. They can perform queries directly on our NoSQL database or process the CSV files that are generated daily (e.g., for more complex analyses on smaller amounts of data). Those experiments can use the whole range of MONROE data, since the moment it started to collect information, and for all the nodes in all the countries, and can be repeated at any point in time. *Active studies* are executed on MONROE nodes via explicit scheduling. They use the experiment services provided as Docker containers and schedule them on real nodes through the platform Web user interface. Those experiments can consist of any software compatible with the container architecture and use all networking resources available in the nodes at the moment of execution, subject to user quotas availability.[14] Experiments can be repeated as desired to verify the consistency of the results or to analyze changes on network behavior along time. The new data generated by active experiments may become part of the dataset available for passive experiments.

Apart from the experiments described in what follows, the MONROE Consortium is in the process of expanding its range of supported measurements and

[14]MONROE users receive a share of the platform resources. Moreover, each MBB interface has limited monthly traffic allowance.

thus enrich the dataset MONROE collects and offers to the community.

3.2.1. Studies by the Consortium

In what follows we describe some of the key studies conducted by using the available MONROE experiment containers, and show samples of our measurement campaigns. However, here we only focus on showcasing the kind of experiments that can be performed and put no emphasis on performance figures and comparisons between services offered by different operators. Therefore, we do not provide a complete and exhaustive set of experiments for all operators and all countries in which we have run the measurements. The results shown in what follows are not representative of the full coverage and service offered by operators across Europe, although the platform could be used to pursue such goal.

Metadata/QoS analysis to build coverage and latency maps. MONROE deployment in public transportation vehicles enables evaluation of MBBs on wide urban mobility environments. Route predictability provides high confidence, whereas measurements taken at similar positions on different hours allow comparing the behavior of the MBBs at different times (e.g., rush hour versus normal hours).

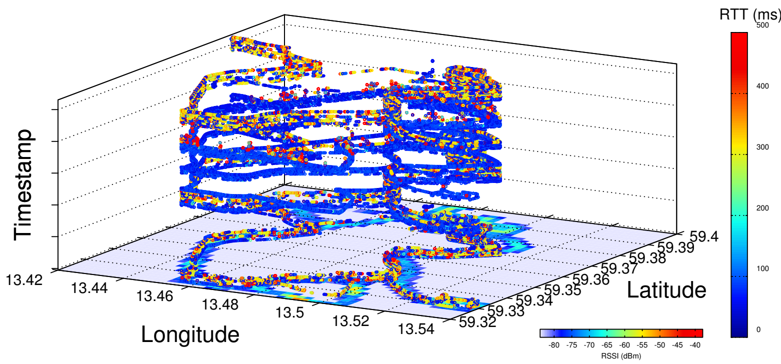
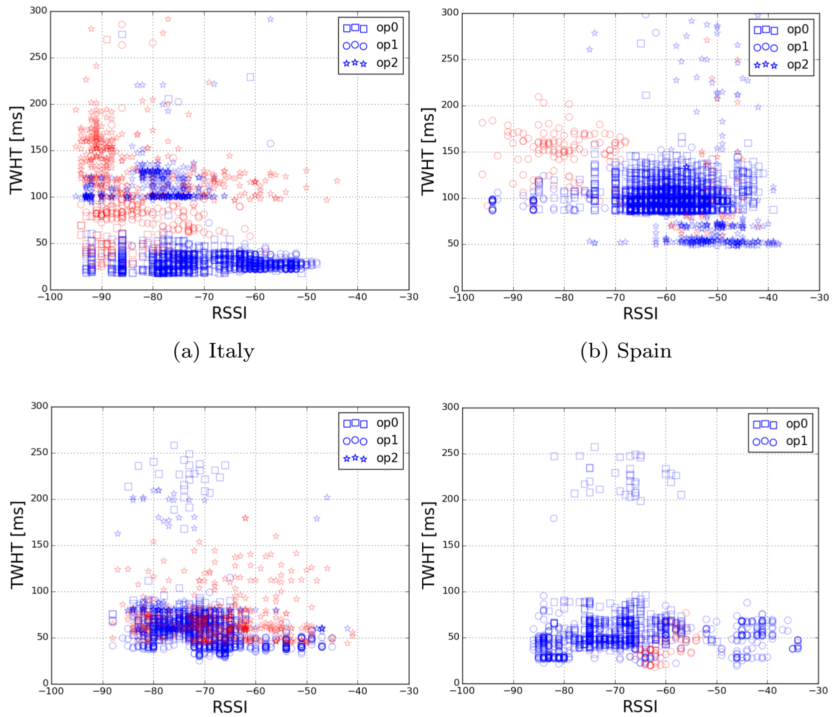


Figure 5: RTT and RSSI measured in a bus at Karlstad, Sweden, over a few observation days. Average RSSI values are shown on the XY plane. Individual RTT measures are plotted on the Z-axis using their relative timestamps as height to visualize successive laps.

Figure 6: This representation of link technology for the bus at Karlstad reveals that 4G coverage is consistently available for the complete route during the analyzed period.

Figure 5 follows the typical route of a bus around Karlstad (Sweden), showing the measured RSSI (signal strength) and RTT (ICMP ping). The different laps along several days are represented vertically ascending to ease the visualization of the dense information obtained. Figure 6 shows the negotiated link technology for the same route. The analysis of the collected data (signal strength, link technology and measured delay) gives insights into the performance perceived by users during their bus trips. Such information might then be used by network operators to improve the service offered to commuters.

Based on the same dataset and on theory and observations that show that fading follows a Rice distribution under line-of-sight conditions, while it follows a Rayleigh distribution otherwise [10], we are currently developing a method



(a) Italy (b) Spain

(c) Sweden (d) Norway

Figure 7: TCP three-way handshake times (TWHT) obtained using the HTTP download experiment for bandwidth measurement with different operators versus the RSSI reported in MONROE metadata. Blue and red correspond to 4G and 3G samples, respectively.

to infer which distribution yields a better fit for experimental data, potentially providing information to operators to optimize the location of base stations.

Traffic analysis and network monitoring with Tstat. We have used Tstat to study the performance of TCP flows as observed by the MONROE nodes. As an example, Figure 7 shows a correlation between three-way handshake time as measured by Tstat, and RSSI from the metadata, illustrating the many possibilities that MONROE creates for cross-domain data analysis.

Operator benchmarking with cross-country performance. MONROE enables comparison of different operators (in terms of network characteristics and user-perceived application performance) in and among countries. For

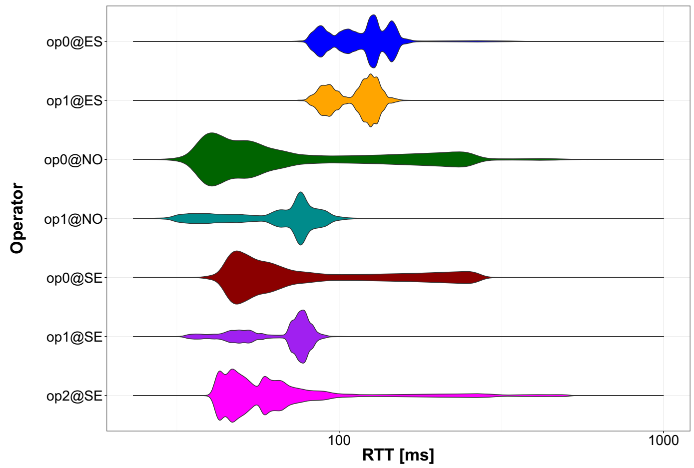
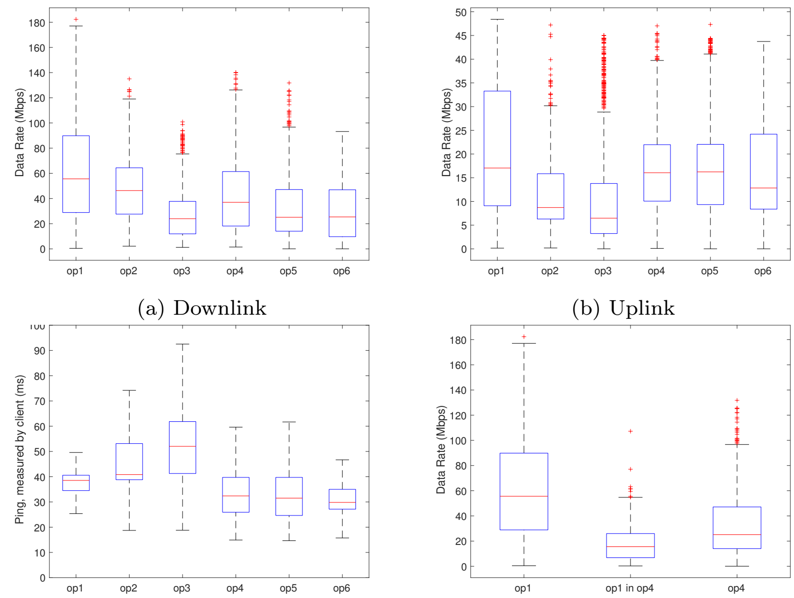


Figure 8: Violin plots of the RTT measurements for different operators in Spain (ES), Norway (NO) and Sweden (SE).

this purpose, multiple MONROE services, such as the ICMP ping container, and the Nettest container can be used.

Figure 8 shows a violin plot for the RTT samples collected (using ICMP ping) during one week with 30 stationary nodes for 7 different operators in 3 countries. Each "violin" shows the probability density of the RTT at different values; the higher the area, the higher the probability of observing a measurement in that range. Nodes in Norway and Sweden exhibit lower delays than nodes in Spain because they are closer to the target measurement server, which is hosted in the MONROE backend in Sweden. Interestingly, measurement variance is much higher than in fixed networks, showing that MBBs introduce complexity even for such basic tests as RTT monitoring. For example, RTT measurements exhibit typically a multimodal distribution that corresponds to the different access delays faced by different radio access technologies (e.g., 3G vs. 4G). MONROE repetitive measurements enable correlation with time, location and context conditions such as variations in signal strength.

It is also possible to benchmark operators using the Nettest container. Running as a base experiment, this container has provided more than 350 thousand measurements over stationary and mobile nodes in Norway and Sweden since



(a) Downlink (b) Uplink (c) Ping (d) Downlink in Roaming

Figure 9: MONROE Nettest base experiment results.

June 2017. Figure 9 presents an overview of the downlink and uplink data rate, as well as latency values for stationary nodes and 6 operators (3 in Sweden, 3 in Norway), including an example case of roaming. For each operator camping on its own network, the Nettest server in the corresponding country has been used.

The roaming example in Figure9 (d) shows the downlink data rate for operator *op1* (Sweden) camping on *op4* (Norway), compared with the native downlink data rates for *op1* and *op4* from Figure9 (a). For this comparison, client nodes in Norway using *op1* SIMs have been used, where the measurements have been conducted against the Nettest server in Norway.

Investigating the speed of mobile broadband. In [11] we present our experience estimating the download speed offered by actual 3G/4G networks. For that experiment, we analyzed data from 50 nodes in 4 countries over 11 operators during more than two months, using the tsat container. The conclusion of that study is that measuring the performance of MBB networks is quite complex as different network configurations such as the presence of NATs or

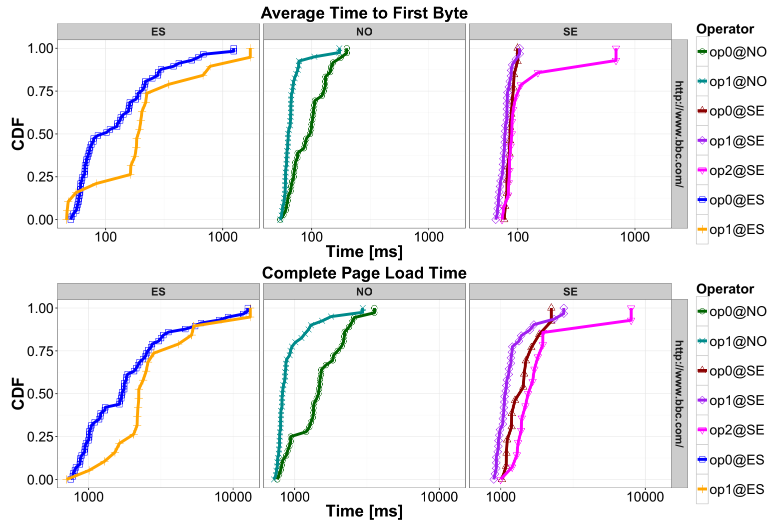


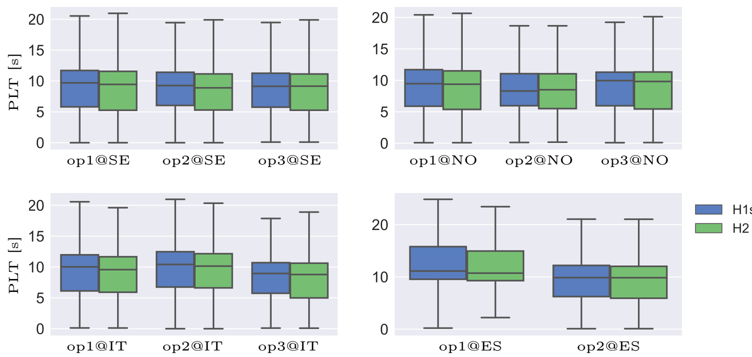
Figure 10: Average Time to First Byte and Complete Page Load Time for some operators in Spain (ES), Norway (NO) and Sweden (SE) for `www.bbc.com`.

PEPs, which do vary over time, have a significant impact on measurements.

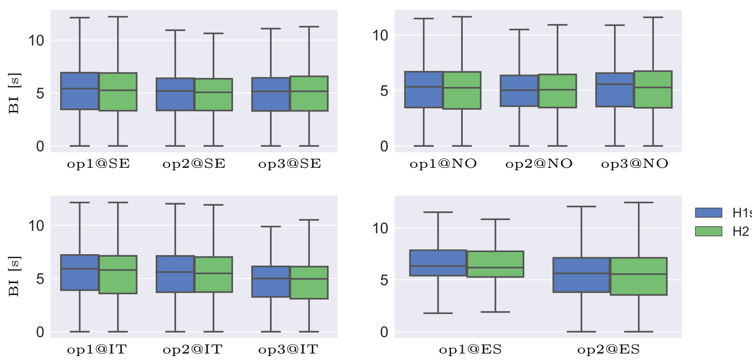
We have made similar observations using the active Nettest container, where the effect of measurement methodology has proven to be a key factor affecting reported data rates. Currently, we have identified 3 main aspects of active measurements that influence data rate as: number of parallel TCP flows, measurement duration, and server location.

Web performance. In [4] we show preliminary results from our experiments on web page load time (PLT) and proxy identification over mobile broadband networks. There, we use a headless browser to fetch two popular websites from 37 nodes operating in four countries and using 11 operators. As an example, we observe large variations of PLT for the same website between Sweden and Norway. In that work we also report results on identification of Performance Enhancing Proxies (PEPs) in MBBs.

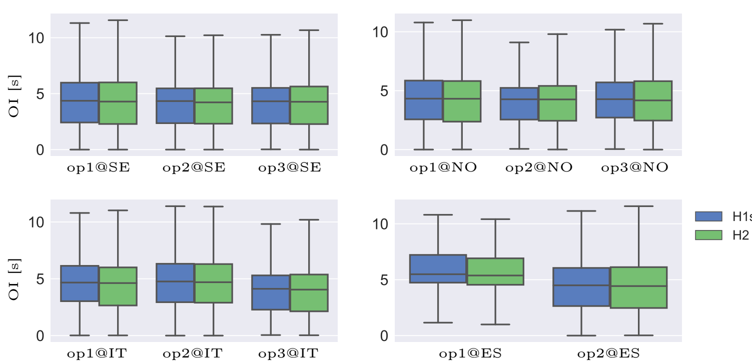
In Figure 10, we present the CDFs of the complete page load time and average time-to-first-byte for `www.bbc.com` broken down per country. Interestingly, for the Spanish operators we detected multiple DNS iterations, which partially account for their higher time-to-first-byte values.



(a) PLT statistics



(b) BI ststistics



(c) OI statistics

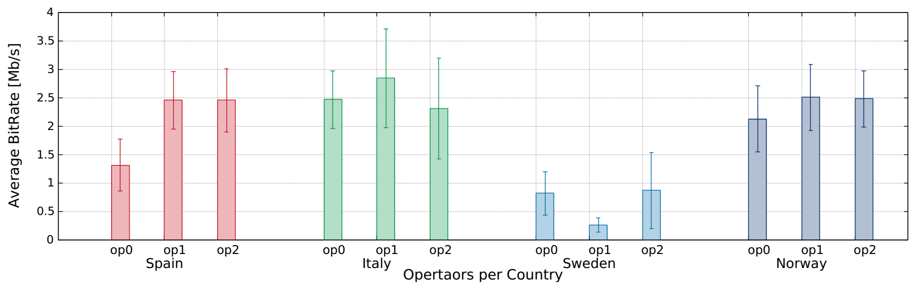
Figure 11: Country-wise overall webpage download operator's performance considering PLT, BI and OI.

If we consider multiple websites, we obtain the results shown in Figure 11. In there, we show not only the PLT metric, but also two time-integral QoE metrics, namely the ObjectIndex (OI) and ByteIndex (BI) metrics defined in [12]. BI and OI give an idea of the speed at which the pages are made available and build up at the reader's browser. Such metrics show that overall Web performance is similar across different countries and operators, with only slight variations. At this aggregate level, we also observe similar performance between HTTP versions (indicated in the figure as H1s in case of version 1.1 with TLS, and H2 in case of version 2.0). This indicates that the protocol version has little impact on performance.

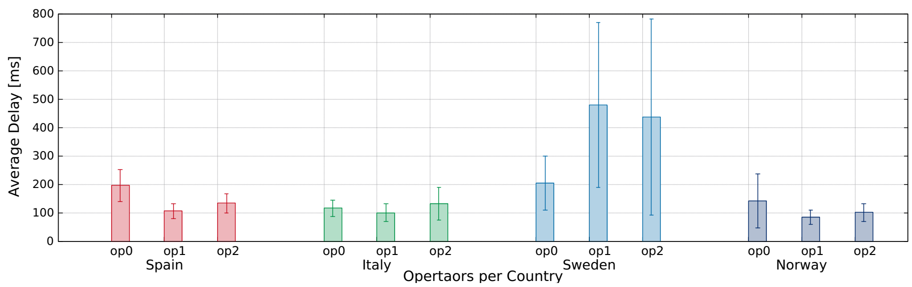
WebRTC performance. We have tested WebRTC services using static and mobile nodes. Specifically, we have connected over HTTPS a Google Chrome WebRTC client from a computer in our lab to MONROE nodes running the WebRTC container. Therefore, the stream goes through the cellular access of the MONROE node, then goes through the Internet and a multi-gigabit connection that connects to our lab. The bottleneck of the WebRTC stream is therefore the MBB network, which the MONROE node connects to.

We use Google Chrome, which offers statistics on peer-to-peer connections, which include WebRTC streams. The resulting logs contain, per each individual stream, the timing and headers of packets received as well as the timing of various internal events such as received frames, losses, bitrate, delay and jitter.

For static WebRTC streamers, we show sample results for multiple operators in four countries in Figure 12, in terms of bitrate and delay. In there, we see that the media stream was smooth in most of the cases, with limited delay and jitter, and bitrates of a the order of a few Mb/s, corresponding to acceptable frame rates of about ten frames per second. However, the results for Swedish operators are not very good, which is in contrast with other observations on the quality offered by those operators. This is an example of experiment that need to be interpreted jointly with metadata. In fact, observing our logs, we have discovered that the SIM cards used for the experiment had simply exhausted their monthly data allowance, which resulted in severe rate limiting experienced



(a) Bitrate.



(b) Delay.

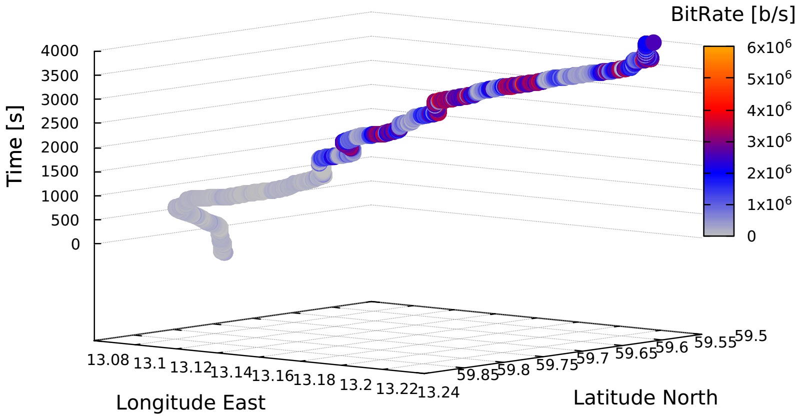
Figure 12: WebRTC performance figures observed for static nodes. Swedish operators were imposing rate limiting due to monthly data usage issues.

by the users.

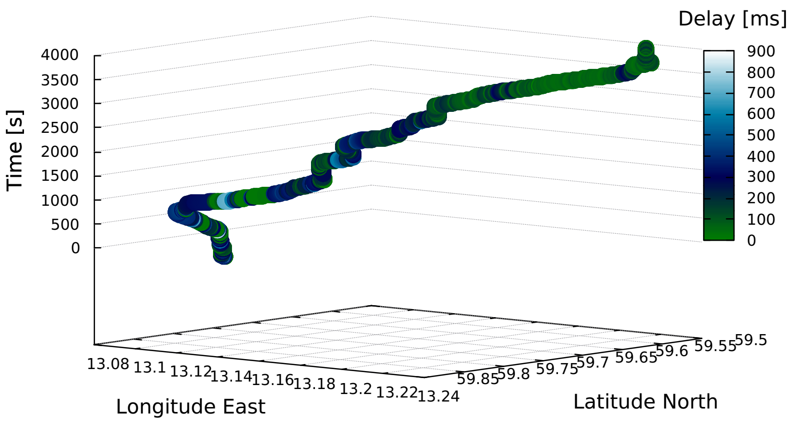
Finally, Figure 13 gives an example of performance for WebRTC with a mobile MONROE node. The figure shows that coverage quality can change a lot over a trip, and so both bitrate and delay suffer large variation. These results point out that current MBB networks might not be ready to fully support WebRTC services on the move.

3.2.2. Studies by external experimenters

Here we give some specific examples of experiments designed by external users and deployed on the MONROE platform. Note that, thanks to the openness of our platform, some of the described experiments have been built on top of MONROE, by extending our nodes with additional hardware and/or software. For detais on extensions and results obtained by experimental researchers, in



(a) Bitrate.



(b) Delay.

Figure 13: WebRTC performance figures observed for a mobile node mounted on a bus.

what follows we give specific pointers on a case-by-case basis.

Software radio extensions. The SOPHIA project has developed an extension to enhance MONROE nodes with software radio capabilities. In [13], its members present detailed performance measurements of LTE networks to illustrate the potential benefits and new possible passive measurements obtained by decoding the control channels of LTE.

Forecasting LTE cell congestion. In [14], the authors try to forecast the average downlink throughput for LTE cells using data collected from multiple MONROE probes and to apply that knowledge to self-organizing network strategies to shift coverage and capacity according to predicted demand. This group updated some MONROE nodes to address the benchmarking of voice calls, showing the flexibility of the platform nodes.

Available Bandwidth measurement on SDN deployments. In [15], the authors employ MONROE as a testbed to study the complexity of available bandwidth estimation using SDN-based active measurements. They conduct their experiments using one node in each of the four main countries of the project. Their ongoing work tries to improve the accuracy and reliability of existing tools, using the MONROE testbed to isolate and better understand different aspects of the measurement process.

Designing application performance with MBB analytics. The authors of [16] use the radio parameters measured by MONROE nodes to determine the best application protocol for a service, identifying the most suitable key performance indicators to characterize the network state. These type of works are very relevant to close the gap between network performance measurements and user experience. Interestingly, the authors see an opportunity on the data generated by other experiments running in the platform (and made openly available by the respective researchers) as a means to obtain additional data points for their own investigation.

Surveying DSCP modifications in mobile networks. MONROE is used by a group of researchers in [17] to conduct a survey on path-level treatment of DiffServ packets in MBB networks and identify behaviors that potentially violate the IETF specifications. DiffServ enables the classification of traffic into QoS classes via usage of the Differentiated Services Code Point (DSCP) field in the IP packet header. Using MONROE to analyze the behavior at the edge mobile network, they find that there is a high probability that the corresponding fields are overwritten in the first two network hops.

Path protocol transparency. *PATHspider* [9] is a tool developed for A/B testing of path transparency. It allows testing the feasibility of deploying new protocols in the Internet and quantifying the impact of path impairments and of middleboxes. In [18], the authors, in collaboration with part of the

MONROE consortium, present the results of adapting PATHspider, to the realm of commercial mobile networks using MONROE nodes deployed by themselves in the UK. Among their conclusions, the most relevant is that MBB networks provide a considerably different environment—and therefore very valuable—with respect to the one provided by the cloud access points that PATHspider was using in the past.

4. Related work

Due to growing interest by regulators, policy makers and networking community, several nationwide efforts to measure the performance of home and mobile broadband networks (e.g., the US FCC's Measuring Broadband America initiative [19]) have been initiated. MONROE goes beyond proposing a trans-national platform.

In contrast with operator-driven measurement campaigns [20, 21, 22], or existing small-case drive-by tests [23], MONROE offers open access to cross-operator collected data, including device-level metadata, which is key to interpret measurement results, across a wide variety of locations.

Moreover, there have been several crowdsourcing projects devoted to measure MBBs using tools such as Mobiperf,[15] Netalyzer [24] and Haystack [25]. Such projects allow crawling through mobile network performance factors to identify the causes of experienced performance figures. In general, such approaches lack rich metadata due to the privacy concerns created by the involvement of real users, hindering the analysis of their datasets. Also, reliance on users can provide high coverage, but at the cost of repeatability regarding location, route or equipment. However, in combination with a platform like MONROE, they could be used in a more systematic and controllable way, as proposed and discussed in [26].

[15] https://sites.google.com/site/mobiperfdev/

5. Conclusions

In this article, we have described the unique EaaS offered by MONROE and discussed how it allows to collect, curate and make available valuable and uniquely rich and open data sets to the community. We have focused on how MONROE helps to improve the knowledge on the usage and behavior of current and future commercial mobile broadband networks. We have also explained the main design characteristics of the platform that make it unique and how, from the generation of data at the nodes to their storage in a NoSQL database that can scale past billions of records, MONROE offers the unprecedented possibility of data analysis across all the nodes and lifespan of the platform. We have presented several and key experiments designed by the MONROE Consortium and by external experimenters. Eventually, to illustrate the potential and flexibility of the platform, we have presented samples of results from our own experiments and from several other groups that have been granted access to our platform.

References

[1] M. P. Quirós, V. Mancuso, V. Comite, A. Lutu, Ö. Alay, S. Alfredsson, J. Karlsson, A. Brunstrom, M. Mellia, A. S. 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 & CHaracterization, WiNTECH@MobiCom, Snowbird, UT, USA, October 20, 2017, 2017, pp. 9–16. doi:10.1145/3131473.3131485.
URL http://doi.acm.org/10.1145/3131473.3131485

[2] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, R. Schatz, YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks, in: European Conference on Networks and Communications (EuCNC), 2015.

[3] O. Alay, A. Lutu, R. García, M. Peón-Quirós, V. Mancuso, T. Hirsch, T. Dély, J. Werme, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunström, A. Safari Khatouni, M. Mellia, M. Ajmone Marsan, R. Monno, H. Lønsethagen,

MONROE, a distributed platform to measure and assess mobile broadband networks: Demo, in: Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental evaluation, and Charaterization, WiNTECH '16, ACM, New York, NY, USA, 2016, pp. 85–86.

[4] O. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunström, A. Safari Khatouni, M. Mellia, M. Ajmone Marsan, Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks, in: Proc. of ACM Mobicom., 2017.

[5] D. Merkel, Docker: Lightweight Linux Containers for Consistent Development and Deployment, Linux J. 2014 (239).

[6] A. Finamore, M. Mellia, M. Meo, M. M. Munafo, P. D. Torino, D. Rossi, Experiences of internet traffic monitoring with tstat, IEEE Network 25 (3) (2011) 8–14.

[7] P. Casas, P. Fiadino, S. Wassermann, S. Traverso, A. D'Alconzo, E. Tego, F. Matera, M. Mellia, Unveiling network and service performance degradation in the wild with mplane, IEEE Communications Magazine 54 (3) (2016) 71–79.

[8] A. Schwind, M. Seufert, O. Alay, P. Casas, P. Tran-Gia, F. Wamser, Concept and Implementation of Video QoE Measurements in a Mobile Broadband Testbed, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[9] I.-R. Learmonth, B. Trammell, M. Kühlewind, G. Fairhurst, PATHspider: A tool for active measurement of path transparency, in: First ACM/IRTF Applied Networking Research Workshop, Berlin, Germany, 2016.

[10] H. Bai, M. Atiquzzaman, Error modeling schemes for fading channels in wireless communications: A survey, IEEE Communications Surveys Tutorials 5 (2) (2003) 2–9.

[11] A. Safari Khatouni, M. Mellia, M. Ajmone Marsan, S. Alfredsson, J. Karlsson, A. Brunström, O. Alay, C. M. A. Lutu, V. Mancuso, Speedtest-like Measurements in 3G/4G Networks: The MONROE Experience, in: Proc. of ITC29, 2017.

[12] E. Bocchi, L. De Cicco, D. Rossi, Measuring the quality of experience of web users, SIGCOMM Comput. Commun. Rev. 46 (4) (2016) 8–13. doi:10.1145/3027947.3027949.

[13] P. Sutton, I. Gomez, MONROE-SOPHIA - A Software Radio Platform for Mobile Network Measurement, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[14] P. Torres, P. Marques, H. Marques, R. Dionísio, T. Alves, L. Pereira, J. Ribeiro, Data Analytics for Forecasting Cell Congestion on LTE Networks, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[15] G. Aceto, V. Persico, A. Pescapé, G. Ventre, SOMETIME: SOftware defined network-based Available Bandwidth MEasuremenT In MONROE, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[16] I. Alepuz, J. Cabrejas, J.-F. Monserrat, A.-G. Perez, G. Pajares, R. Gimenez, Use of Mobile Network Analytics for Application Performance Design, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[17] A. Custura, A. Venne, G. Fairhurst, Exploring DSCP modification pathologies in mobile edge networks, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[18] I.-R. Learmonth, A. Lutu, G. Fairhurst, D. Ros, O. Alay, Path Transparency Measurements from the Mobile Edge with PATHspider, in: Proc. of the IEEE/IFIP Workshop on Mobile Network Measurement, 2017.

[19] FCC, 2013 Measuring Broadband America February Report, Tech. rep., FCC's Office of Engineering and Technology and Consumer and Governmental Affairs Bureau (2013).

[20] E. Halepovic, J. Pang, O. Spatscheck, Can you GET me now?: Estimating the time-to-first-byte of HTTP transactions with passive measurements., in: Proc. of IMC, 2012.

[21] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, J. Wang, A first look at cellular network performance during crowded events, in: Proc. of SIGMETRICS, 2013.

[22] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z.-M. Mao, S. Sen, O. Spatscheck, An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance, in: Proc. of SIGCOMM, 2013.

[23] Tektronix, Reduce Drive Test Costs and Increase Effectiveness of 3G Network Optimization, Tech. rep., Tektronix Comm. (2009).

[24] C. Kreibich, N. Weaver, B. Nechaev, V. Paxson, Netalyzr: Illuminating the edge network, in: Proc. of the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 246–259.

[25] N. Vallina-Rodriguez, Illuminating the third party mobile ecosystem with the lumen privacy monitor, in: FTC PrivacyCon 2017, 2017.

[26] M. R. Fida, A. Lutu, M. K. Marina, O. Alay, Zipweave: Towards efficient and reliable measurement based mobile coverage maps, in: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9. doi:10.1109/INFOCOM.2017.8057098.