Data-Driven Emulation of Mobile Access Networks

Data-Driven Emulation of Mobile Access Networks

Ali Safari Khatouni†, Martino Trevisan‡, Danilo Giordano ‡
†Dalhousie University, ali.safari@dal.ca
‡Politecnico di Torino, first.last@polito.it

***Abstract*—Network monitoring is fundamental to understand network evolution and behavior. However, monitoring studies have the main limitation of running new experiments when the phenomenon under analysis is over e.g., congestion. To overcome this limitation, network emulation is of vital importance for network testing and research experiments either in wired and mobile networks. When it comes to mobile networks, the variety of technical characteristics, coupled with the opaque network configurations, make realistic network emulation a challenging task.**

In this paper, we address this issue leveraging a large scale dataset composed of 500M network latency measurements in Mobile BroadBand networks. By using this dataset, we create 51 different network latency profiles based on the Mobile BroadBand operator, the radio access technology and signal strength. These profiles are then processed to make them compatible with the `tc-netem` emulation tool. Finally, we show that, despite the limitation of current `tc-netem` emulation tool, Generative Adversarial Networks are a promising solution used to create realistic temporal emulation.

We believe that this work could be the first step toward a comprehensive data-driven network emulation. For this, we make our profiles and codes available to foster further studies in these directions.

I. Introduction

Mobile devices dramatically revolutionized the way we interact with the Internet, offering access to the web, video, and messaging applications in mobility with a capacity comparable with the wired network [1]. As a result, nowadays a huge amount of the internet traffic is generated in mobility.

In this scenario, Mobile BroadBand (MBB) network operators have to face severe challenges to catch up with the rapid evolution of the network condition, location, and technologies.

For these reasons *monitoring* the network became a fundamental task needed to understand the network behavior. By monitoring and measuring, researchers can quantify the evolution of technologies, protocols, setups and website design. Not surprisingly over the years, the research community has put a lot of effort into measuring the benefits of new technologies [2], [3], [4]. Previous studies focused on different aspects, and often addressing specific angles of this complex ecosystem [5], [6], [7], [8], [9], [10].

However, despite monitoring is a perfect solution to profile and understand specific measurements, e.g., network latency, it does not allow *per se* to run repeatable experiments. Once a particular phenomenon, e.g., congestion, is over, running new experiments on the same condition is impossible. To overcome this limitation, a specific phenomenon can be modeled, and then the model can be used in an *emulated* environment to run new experiments with the same condition. As a result, new experiments can be performed without the needs of a particular phenomenon happening.

In this work, we address this problem by using an open dataset collected by the MONROE platform[1] The MONROE platform consists of 96 nodes in four European countries. Each node is equipped with MBB subscription that continuously (i.e., every 1 second) measures network latency towards dedicated servers by using ICMP `pings`. In addition to network latency, the MONROE nodes collect different physical-layer information such as frequency, radio access technology (RAT), signal strength, etc.

By using this dataset, we first create MBB network latency profiles separately for each operator, RAT (i.e., 3G and 4G), signal strength quality, and roaming condition. By leveraging more than 500 M latency measurements from 11 different service providers, in 96 different locations, from 4 different countries, we created 51 latency profiles. Secondly, we feed each profile into the `tc-netem` Linux tool[2] creating an emulated environment where we can run further experiments under specific network latency conditions, e.g., evaluating the users' experience accessing the network with the poor 4G connection.

Finally, we explore more sophisticated techniques to achieve a realistic temporal emulation. We extract time series from latency measurement, and use them to train Generative Adversarial Neural Network (GAN) [11] models. GANs have been proved to be powerful generative models, able to understand complex distributions behind e.g., images and text. We show that GANs can model the time dimension of latency measurement, generating e.g., peaks, following the same latency distribution of the training data i.e., offering realistic profiles. Despite the promising capabilities of GANs, the *de facto* standard tool (`tc-netem`) lacks the capability of using user-defined time series.

Our work poses itself as preliminary work to emphasize the challenges and the importance of latency profiling and emulation in operational mobile networks. We make our `tc-netem` distribution tables available to the community [12] to allow other researchers to replicate our results and practitioners to benefit from this data.

The rest of this paper is organized as follows. In Section II, we briefly discuss the related work. In Section III, we describe the measurement setup in which the data is collected, while in Section IV we discuss our methodology to generate latency

[1]https://www.monroe-project.eu/access-monroe-platform/
[2]http://man7.org/linux/man-pages/man8/tc-netem.8.html

profiles. In Section V, we present time series generation using GANs. Finally, Section VI concludes the paper and discusses future work.

II. Related Work

Monitoring performance is fundamental for regulators to ensure transparency and the general quality level of the basic Internet access service [13]. Some of them responded to this issue with ongoing nationwide efforts [14]. Continuous measurements are crucial to understanding the latency and performance of mobile networks. Network latency plays a key role when it comes to the users' Quality of Experience (QoE) [15], [16]. Balachandran *et al.* [17] highlight the impact of low-level measurable radio network characteristics on the user QoE during web browsing. A survey of end-to-end delay prediction methods is offered by authors of [18]. They show several methodologies to estimate latency, e.g., *Queuing Network Modelling*, *System Identification*, *Time Series Approach*, and *Neural Networks*. Nunes *et al.* [19] focus on TCP performance in a Mobile Ad Hoc network and propose a machine learning technique called *Experts Framework*. Mandalari *et al.* [20] show that a roaming user in Europe suffers an additional latency of ∼60 ms or more, depending on geographical distance. Safari *et al.* [21] illustrate accurate mobile network latency prediction is not effective by using the common machine learning algorithms. However, while performing monitoring studies the problem of online privacy should not be overlooked. Many parties can collect and access these kind of data other than researchers. In [22], the authors highlight what are the privacy and ethical issues that arise for users, companies, scientists and governments and presenting the current legislation.

Network emulation is of crucial importance for network testing, and it is a well-studied topic in the literature. The Linux operating system includes `tc-netem`, a tool for this purpose since more than 20 years. Considering emulation of *mobile* networks, many works aimed at achieving realism with fine-grained models of network devices. For instance, NIST Net [23] is a flexible and powerful emulator for WANs, while authors of [24] exploit advanced modeling methods to build a mobile network emulation environment. More recently, mobile network emulation followed the general trends of research in networking, and has been used to mimic real deployments of e.g., Software Defined Networks [25] or flying vehicles networks [26]. There are few research studies building on a data-driven approach to obtain a realistic emulation. Seminal work [27] proposes a pure trace-driven network emulation that re-creates the observed end-to-end characteristics of a real wireless network. More recently, KauNet [28] provides pattern-based emulation with a higher level of detail, controlling the behavior of each individual packet. In contrast to previous works, we build on a large measurement dataset of operational mobile networks and make our latency profiles available to the community and compatible with the `tc-netem`.

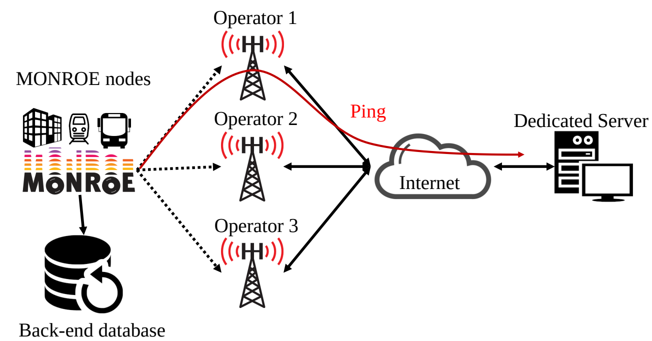


Fig. 1: The experimental setup for `ping` measurement in the MONROE platform.

Machine learning has not yet been used in the context of mobile network emulation, but the recent advances in neural networks make it a promising technique for modeling complex phenomena. Recently, GANs have been proposed as generative models. Introduced in 2014 [11], GANs are used for image generation [29] and recognition [30], text-to-photo synthesis [31] and URL generation [32]. In this paper, we show that GANs are able to model MBB network latency, even though the real applicability of this approach is limited by inflexibility of the current *de facto* standard tool, namely `tc-netem`.

III. Measurements Setup

In this section, we briefly describe the measurement setup and the employed dataset.

A. Measurement Infrastructure

Systematic repeatable measurements are crucial for evaluating network performance and assessing the quality experienced by end-users. As such, researchers have built platforms dedicated to broadband networks, e.g., RIPE Atlas [3], CAIDA Ark [4], or PlanetLab [5]. In contrast to them, MONROE [33] is a unique platform that enables controlled experimentation with different commercial mobile carriers. It enables users to run custom experiments and to schedule experimental campaigns to collect data from operational MBB and Wi-Fi networks, together with full context information (metadata). It covers 4 countries in Europe (Italy, Norway, Spain, and Sweden) with more than 100 nodes equipped with Ethernet, WIFI and 3G/4G interfaces with commercial subscriptions.

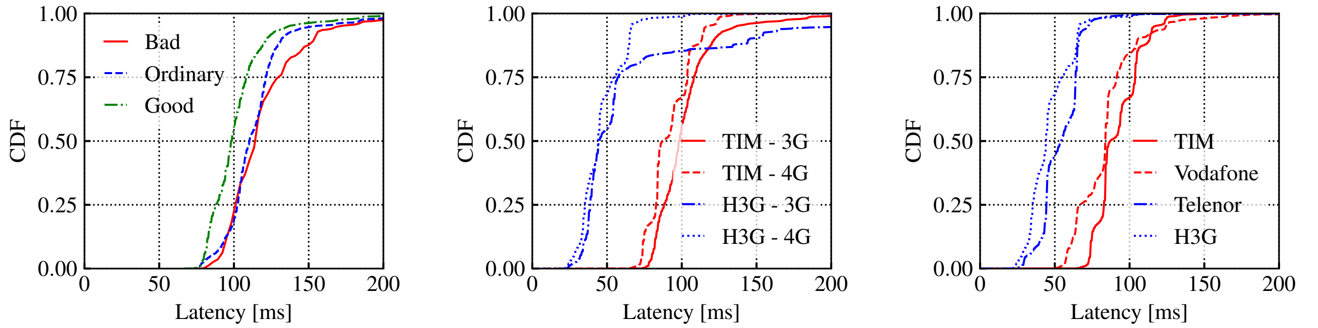
The MONROE platform allows us to access the information about network, time and location of experiments, as well as metadata from the mobile modems, including, e.g., signal strength, RAT, cell identifier for each network provider[6]. Each node performs a set of predefined experiments (e.g., `ping`,

[3] https://atlas.ripe.net/

[4] http://www.caida.org/projects/ark/

[5] https://www.planet-lab.org/

[6] The description and detail information about metadata is available on https://github.com/MONROE-PROJECT/UserManual



(a) An example of different signal quality for TIM 3G.

(b) An example of good 3G vs. good 4G for TIM and H3G.

(c) Some samples of the good 4G. for different operators.

Fig. 2: ECDFs for different operators and scenarios.

TABLE I: Statistics on the dataset collected during 2018 by the MONROE platform.

Country	Nodes	Operators	Latency samples
Italy	27	3	112.2 M
Norway	18	3	122.4 M
Spain	19	2	80.4 M
Sweden	32	3	191.4 M
Total	96	11	506.4 M

passive monitoring, etc.) continuously. Experiment results are collected in the central database, and stored in a Hadoop based cluster. In the following, we provide details about the experimental setup and describe the considered metrics.

Figure 1 shows the experiment setup we consider in this paper. The leftmost element is the MONROE node. It is equipped with three different mobile modems, each equipped with a SIM card of a different operator. Traffic (e.g., `ping`– red curve) flows through the specified operator and the Internet, toward the selected server – on the rightmost part of the figure. This allows us to measure up to three operators at the same location and time, and using identical device and software, thus, limiting potential sources of bias. This allows one to understand the coverage, and all the time-varying parameters that may impact the latency.

ICMP `ping` (84 Byte payload) experiments run regularly every second to measure the latency (round-trip time) from nodes toward a dedicated server. The system logs collected data to the central repository after the experiments. Beside latency, we analyze a set of meta-data that characterize the specific context in which each experiments run. Especially, we consider access network parameters: They include parameters from the physical layer such as radio status during the experiment (RSSI, RSRQ, and RSRP).

B. Dataset Description

In this work, we present the data collected on the MONROE platform during the whole year of 2018. It contains measurements from 96 MONROE nodes, measuring 11 operators in 4 European countries, i.e., Italy, Norway, Spain, and Sweden. Nodes operate in a different scenario such as university campuses, urban areas, countryside, aboard buses, and trains. This dataset focuses on latency measurements. In total, we collected latency for about 500 M samples, as detailed in Table I.

TABLE II: Binning boundaries for 3G and 4G. Values are expressed in dB.

Quality	3G	4G
Poor	$rssi \leq -100$	$rssi \leq -85$
Ordinary	$-100 < rssi \leq -85$	$-85 < rssi \leq -75$
Good	$rssi > -85$	$rssi > -75$

IV. Emulating Mobile Access Network Latency

In this section, we describe the proposed approach to emulate mobile access network latency.

A. Data processing and cleaning

Provided the dataset described in the previous section, we process it to extract clean latency profiles for scenarios that differ for the SIM card operator, RAT, etc. Given the large size of data, all processing is done using Apache Spark for scalability reasons.

We first join the `ping` samples with physical layer information. As such, separately for each node, we join the two datasets on the time dimension. We then get rid of those time intervals where few or no `ping` sample is observed due to e.g., network failures or power outages. At this point, we sort `ping` samples by time and extract sequences of fixed length $l = 60$.[7] We then exclude those sequences where physical layer conditions are not stable due to e.g., intra-RAT or inter-RAT handovers, strong variations in signal strength, etc. This step is necessary to obtain clean data and allows us

[7]The node sends a `ping` every second. As such sequences are 60 seconds long.

to remove most of the measurement artifacts. We finally group the obtained sequences to build the sample set composing each latency profile. We group sequences according to:

- **SIM card operator**
- **Radio Access Technology:** 3G or 4G
- **Signal Quality:** we binned signal strength values to three levels, poor, ordinary and good. Binning boundaries are reported in Table II. However, very few samples are observed with *low* signal quality and thus few profiles are finally included.
- **Roaming:** whether the SIM card was roaming on another a country at the moment. Remind that the widespread deployment of *Home Routing* by operator significantly increases the latency of this configuration [20].

This said, we obtain 51 latency profiles for which we observe more than $10\,000$ samples. They include data from 4 countries and 11 operators for a total of more than 500 M `ping` samples. They also differ for the observed signal quality, for which we have 2 profiles with *bad*, 22 with *ordinary* and 27 with *good* quality. Considering RAT, we have 16 profiles for 3G and 35 for 4G, and, overall, 15 are collected while SIM was roaming. Figure 2 shows three examples of latency profiles. Results report the Empirical Cumulative Distribution Function (ECDF) of the latency. The x-axis in each plot of Figure 2 gives the latency in ms and the y-axis gives the probability of the latency being less than the x-axis value. Figure 2a indicates the latency distributions for three classes of signal quality for 3G technology using TIM operator (Italy). There is clear separation for good signal quality (i.e., lower latency) but the latency distributions for ordinary and bad qualities have some similarities. Figure 2b shows the latency distribution for 3G and 4G of TIM (Italy) and H3G (Sweden). There is a clear difference between the two countries for the same scenario (i.e., same technology and same signal quality). Obviously, the 4G experiences lower latency. Figure 2c presents the differences between four operators using 4G with good signal quality. These visual presentations show some examples of generated profiles for multiple operators and technologies.

B. Latency Emulation

We use the obtained latency samples together with the `tc-netem` Linux tool to impose the selected latency profile on the desired network interfaces. We first create a `netem` delay distribution table for each profile. Besides, we compute average and standard deviation for further use, as `netem` requires delay tables to be normalized with respect to such metrics. We then use the delay tables to enforce an latency profile on the selected network interface: the delay table is first copied on the proper system directory, and we then add a `netem` queuing using the corresponding table, and with the values of average and standard deviation computed before. Initial verification confirms that `netem` correctly handles our latency table and the resulting latency is perfectly consistent with the expected outcome. The obtained latency profiles are made public and available to researchers and practitioners [12]. We provide each profile in the form of a custom `netem` delay distribution. We also implemented simple convenience scripts that allow the user to easily install such profiles in the proper system directory and start delay emulation on a selected interface.

V. Experiments using Generative Adversarial Networks

Our approach based on the mere sample distribution has drawbacks that limit the realism of the emulation. Whereas the overall distribution of the emulated and real samples is the same, the approach does not model the time dimension. As such, there is no temporal correlation among the generated samples, and we cannot model time-based events, such as a temporary increase in the latency caused by a short-term network failure.

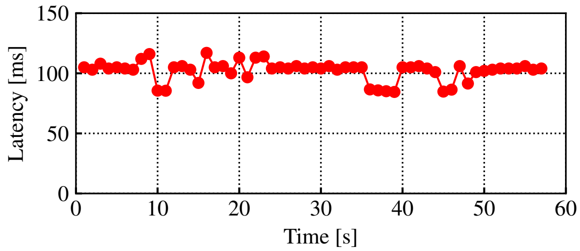
To overcome such limitation, we propose the use of GANs, that have been proven able to model (and generate) complex data such as images [29] and network data [32]. A GAN is composed of a *generator* and a *discriminator* model. Provided with random noise and feedback from the discriminator, the generator task is to produce samples that are similar to the real data, trying to make up artificial samples that cannot be distinguished by the discriminator. The discriminator task instead is to distinguish between real samples in the training set and the artificial ones produced by the generator. As the two models compete to win their adversarial tasks, artificial samples become more and more realistic, whereas the discriminator becomes robust to noise.

In our experiments, we train a GAN for each latency profiles. Figure 4 sketches the building blocks of our GAN during training. We feed them with sequences of length $l = 60$ of latency samples. In the original data, Pings are performed each second, and we remove all sequences with missing data to obtain a fully-formed time series. We tested different architectures for both generator and discriminator models, and we obtained the best results using two stacked LSTMs [34] in both. Examples of real and generated sequences are depicted in Figure 3. The two generated sequences (right) correctly show temporal correlation across samples, and we can identify short-term peaks in which latency increases. Moreover, the generated samples have absolute values consistent with the original, as shown in Figure 5 for two examples. Real and generated distributions mostly overlap, even if they do not perfectly match. Indeed, Kolmogorov–Smirnov tests provide KS statistics in the order of $[0.1, 0.4]$ for the considered profiles.

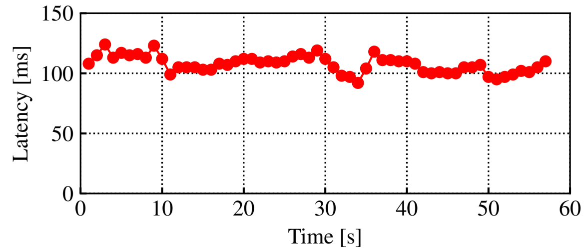
Our results show that using GANs to emulate latency profiles is a promising approach, but technical issues limit the applicability to real cases. In particular, widespread network emulation tools (e.g., `netem`) cannot emulate time-series based latency profiles, and as such require the design and implementation of *ad-hoc* tools.

VI. Conclusion and future work

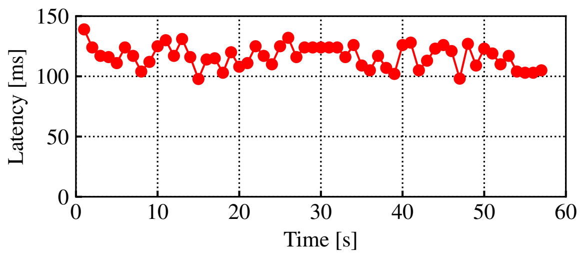
Mobile devices have become one of the main tool used for accessing the Internet and web services. However, the diversity



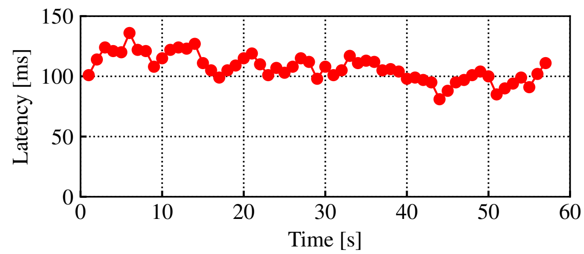
(a) Real latency sequences.



(b) Generated latency sequences.



(c) Real latency sequences.



(d) Generated latency sequences.

Fig. 3: Examples of real (left) and generated (right) latency sequences for the TIM, ordinary 3G profile.

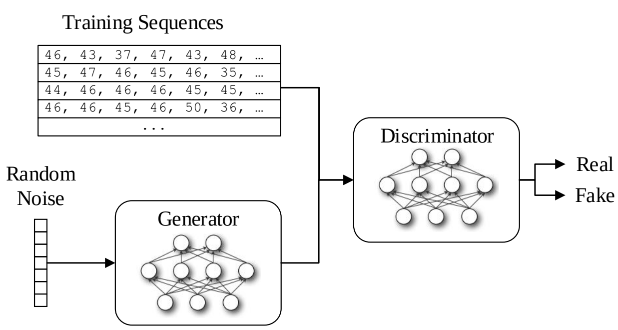


Fig. 4: The architecture of the deployed GANs. They are fed with `ping` sequences of fixed length, i.e., 60 second.

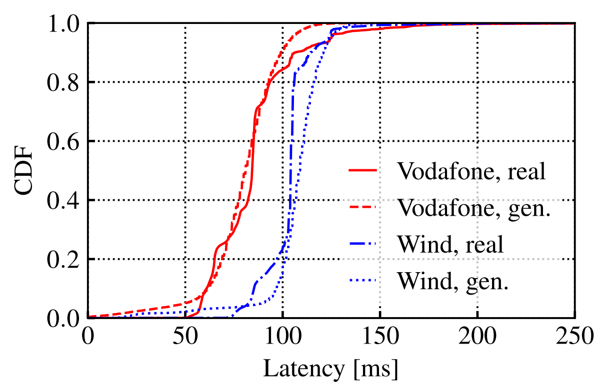


Fig. 5: An example of ECDFs for real and generated samples (good 4G).

of devices, technologies, and mobility makes it highly dynamic and utterly difficult to predict. As such, network emulation is of fundamental importance for testing new applications, new protocols, or performing research experiments.

In this paper, we exploit a large and one-of-the-kind dataset collected in 4 countries from 11 operational operators. We presented two different approaches for emulating MBB network latency. Firstly, we obtained 51 latency profiles, for different operators, RAT, and signal quality. They can be used in `tc-netem` as distribution tables and they are available to the community. Secondly, we explored to use GANs to generate realistic latency time series and found that they fit for the purpose.

This work is the first step toward a comprehensive data-driven network emulation. Firstly, we created latency profiles, as our experiments do not yet include bandwidth and packet loss measurements. Secondly, GANs are promising as generative models to achieve high realism, but a big effort is required to fit them in the current network emulation tools.

Acknowledgements

The research leading to these results has been funded by the European Union's Horizon 2020 research and innovation program under grant agreement No. 644399 (MONROE) and the Smart-Data@PoliTO center for Big Data technologies.

References

[1] L. Vassio, I. Drago, M. Mellia, Z. B. Houidi, and M. L. Lamali, "You, the web, and your device: Longitudinal characterization of browsing

habits," *ACM Transactions on the Web (TWEB)*, vol. 12, no. 4, p. 24, 2018.

[2] J. Sommers and P. Barford, "Cell vs. wifi: on the performance of metro area mobile connections," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, pp. 301–314, ACM, 2012.

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

[4] O. Fagbohun, "Comparative studies on 3g,4g and 5g wireless technology," vol. 9, pp. 133–139, 01 2014.

[5] U. Goel, M. Steiner, M. P. Wittie, M. Flack, and S. Ludin, "Http/2 performance in cellular networks," in *ACM MobiCom*, 2016.

[6] D. Baltrunas, A. Elmokashfi, A. Kvalbein, and O. Alay, "Investigating packet loss in mobile broadband networks under mobility," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 225–233, May 2016.

[7] M. Trevisan, I. Drago, and M. Mellia, "Impact of access speed on adaptive video streaming quality: A passive perspective," in *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pp. 7–12, ACM, 2016.

[8] M. Trevisan, I. Drago, and M. Mellia, "Pain: A passive web performance indicator for isps," *Computer Networks*, vol. 149, pp. 115–126, 2019.

[9] L. Vassio, D. Giordano, M. Trevisan, M. Mellia, and A. P. C. da Silva, "Users' fingerprinting techniques from tcp traffic," in *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pp. 49–54, ACM, 2017.

[10] D. Giordano, S. Traverso, L. Grimaudo, M. Mellia, E. Baralis, A. Tongaonkar, and S. Saha, "Youlighter: An unsupervised methodology to unveil youtube cdn changes," in *2015 27th International Teletraffic Congress*, pp. 19–27, Sep. 2015.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[12] M. Trevisan, A. Safari Khatouni, and D. Giordano, "https://github.com/marty90/mobile-latency-emulator."

[13] networld2020, "Service level awareness and open multi-service internetworking - principles and potentials of an evolved internet ecosystem," 2016.

[14] FCC, "2013 Measuring Broadband America February Report," tech. rep., FCC's Office of Engineering and Technology and Consumer and Governmental Affairs Bureau, 2013.

[15] S. Souders, "The performance of web applications – one-second wonders for winning or losing pcustomers.." https://goo.gl/ErNLcm, Nov 2008. [Online; accessed 07-Aug-2018].

[16] E. S. (Bing) and J. B. (Google), "Performance related changes and their user impact." https://goo.gl/hCr7ka, Jul 2009. [Online; accessed 07-Aug-2018].

[17] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, and H. Yan, "Modeling web-quality of experience on cellular networks," in *Proceedings of the 20th annual international conference on Mobile computing and networking, MobiCom'14*, pp. 213–224, ACM, 2014.

[21] A. S. Khatouni, F. Soro, and 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)*, pp. 71–74, April 2019.

[18] M. Yang, X. R. Li, H. Chen, and N. S. Rao, "Predicting internet end-to-end delay: an overview," in *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, IEEE, 2004.

[19] B. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning approach to end-to-end rtt estimation and its application to tcp," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pp. 1–6, IEEE, 2011.

[20] A. M. Mandalari, A. Lutu, A. Custura, A. Safari Khatouni, O. Alay, M. Bagnulo, V. Bajpai, A. Brunstrom, J. Ott, M. Mellia, and G. Fairhurst, "Experience: Implications of roaming in europe," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), pp. 179–189, ACM, 2018.

[22] L. Vassio, H. Metwalley, and D. Giordano, "The exploitation of web navigation data: Ethical issues and alternative scenarios," in *Blurring the Boundaries Through Digital Innovation*, pp. 119–129, Springer, 2016.

[23] M. Carson and D. Santay, "Nist net: a linux-based network emulation tool," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 111–126, 2003.

[24] N. Ivanic, B. Rivera, and B. Adamson, "Mobile ad hoc network emulation environment," in *MILCOM 2009-2009 IEEE Military Communications Conference*, pp. 1–6, IEEE, 2009.

[25] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 384–389, IEEE, 2015.

[26] S. Rosati, K. Kruželecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, 2015.

[27] B. D. Noble, M. Satyanarayanan, G. T. Nguyen, and R. H. Katz, "Trace-based mobile network emulation," in *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 51–61, ACM, 1997.

[28] J. Garcia, E. Conchon, T. Pérennou, and A. Brunstrom, "Kaunet: improving reproducibility for wireless and mobile research," in *MobiEval 2007*, pp. p–21, 2007.

[29] E. L. Denton, S. Chintala, R. Fergus, *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, pp. 1486–1494, 2015.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint*, 2017.

[32] M. Trevisan and I. Drago, "Robust url classification with generative adversarial networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 3, pp. 143–146, 2019.

[33] 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, and M. Ajmone Marsan, "Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks," *Proc. ACM MobiCom '17*, pp. 70–78, 2017.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.