SOMETIME: SOftware defined network-based Available Bandwidth MEasuremenT In MONROE

Giuseppe Aceto*,◇, Valerio Persico*,◇, Antonio Pescapé*,◇, Giorgio Ventre*,◇
*University of Napoli Federico II (Italy) and ◇NM2 srl (Italy)
giuseppe.aceto@unina.it, valerio.persico@unina.it, pescape@unina.it, giorgio@unina.it

***Abstract*—Mobile Broadband (MBB) access networks are becoming more and more used worldwide, and the devices adopted to access them are increasing in number and complexity (smartphones, mobile hotspots, vehicular infotainment systems). The highly dynamic nature of such scenarios calls for continuous monitoring and measurement of the network, and possibly cross-layer management of network applications. A recent shift in network management, Software-Defined Networking (SDN), is a promising tool to manage such evolved scenario, characterized by constraints due to HW, virtualization, and data plans. In this paper, we present the fundamental ideas and the first findings that underpin the SOMETIME research project, that aims at implementing active measurements leveraging the features provided by SDN technologies. Several platforms and tools are being presented to investigate separately MBB and SDN: we consider as a reference testbed the MONROE platform, a system offering in-the-field MBB experimenting facilities. We adopt MONROE as an use case to highlight the main issues and challenges raised by the SOMETIME vision, investigating the feasibility of SDN-based active measurements for MBB. In more details, we assess the impact of SDN on performance of active measurements, namely Available Bandwidth (ABw) estimation, an end-to-end network metric characterizing the spare capacity on a path. We also report preliminary results on achievable throughput as a first root-cause analysis for poor performance in estimating ABw in MBB scenarios. The preliminary results confirm the expected difficulties in ABw estimation over MBB but also validate the feasibility of SDN-based approaches and suggest future directions for SDN-based enhancement of ABw estimation.**

I. Introduction

Nowadays a mobile terminal is a complex system, hosting multiple network applications, each with different requirements of QoS, besides possibly audio and video traffic and background OS-related control traffic. Moreover, mobile broadband (MBB) access networks are often shared among different devices by means of multi-access mobile devices (called *mobile wireless router*, *mobile hotspot*, *Mi-Fi*) that perform routing and NAT for a network of wireless-connected devices towards an uplink cellular data connection. Finally, vehicles themselves have become the more and more equipped with network applications for different goals (and more can be envisioned) including entertainment, traveling assistance (maps, navigator, traffic news, travel planning), comfort (clima control, seats settings), and maintenance. A mobile broadband measurement procedure that does not account for such sharing of communication resources would have inherent limitations by design. Hence the necessity to experiment in realistic scenarios taking into account multiple concurrent communications in a variety of controlled field conditions, in order to avoid research results that are too narrow in scope or lack real applicability.

Such scenario with multiple communications can be implemented by adopting a Software-Defined Networking (SDN) approach, where the network control plane is logically centralized and the control protocol is abstracted and standardized. More specifically, SDN applied to MBB setups offers networks hosted by a vehicle managed by a (local, remote, or hierarchical local-and-remote) SDN controller, where isolation can be enforced while redundancy can be efficiently exploited. Besides being a very promising approach with ongoing lively research, SDN provides both high flexibility and standardization, ideal for network measurement studies.

One of the most useful metrics for network performance is the end-to-end Available Bandwidth (ABw), that can be considered as the maximum rate that a new packet flow can impose on a path without affecting the other flows sharing part of the path (*cross traffic*).

For the aforementioned reasons we consider the implementation of ABw estimation techniques in an SDN a significant advance of the state of the art in wireless network measurements. The MONROE[1] testbed [5] has been designed purposely to experiment with MBB access networks, thus provides the suitable infrastructure to implement and evaluate a prototype of the measurement system we devised. This led us to design the SOMETIME project (SOftware defined network-based available Bandwidth MEasuremenT In MONROE), accepted in the 1^{st} MONROE Open Call, leveraging the MONROE testbed to perform ABw estimation in an SDN environment from MBB nodes.

In this paper we report the fundamental ideas of the SOMETIME research project as well as the results of preliminary experiments we have conducted on a prototype. A background on the concepts involved in the design and experimenting is described in Section II. The SOMETIME research plans is provided in Section III. The experimental analysis of available alternatives for the ABw estimation tools is reported in Section IV. The setup of the emulation testbed, the experimentation, and the analysis of the results are reported in Section V. A preliminary analysis involving the MONROE testbed is detailed in Section VI.

II. Background

Estimation of ABw in an SDN setup for MBB networks has a huge application potential, but such experimental scenario on the other side poses a number of challenges that have to be carefully considered and addressed in the design of

[1]The MONROE project [1] is an European Union's Horizon 2020 funded research project, aimed building and operating a large-scale experimental platform, targeting MBB and WiFi networks, distributed over multiple European countries.

the experiments and in the deployment of the experimental setup. In the following sections we will introduce the main concepts related with the SOMETIME project, motivating the experiments.

A. Available Bandwidth estimation

The available bandwidth is first defined on each link of a network path. The *available bandwidth* in the time interval $(t-\tau, t)$ for the i-th link, with capacity C_i, is

$$A_i(t-\tau, t) \equiv \frac{1}{\tau}\int_{t-\tau}^{t} C_i(1-u_i(x))dx \tag{1}$$
$$= C_i(1-\bar{u_i}(t-\tau, t)) \tag{2}$$

where τ is the *averaging timescale* and $\bar{u_i}(t-\tau, t)$ is the average utilization of link i during τ. In other words the available bandwidth of a link is the average of unused capacity during the considered time interval. The available bandwidth on a path is defined as the minimum value of available bandwidth of the links composing the path.

The assumptions behind these definitions are: (i) the path is fixed and unique (not subject to routing changes or multipath forwarding); (ii) the routers operate according to FIFO discipline; (ii) the capacity of each link is constant. These assumptions are rarely verified when wireless links are included in the path [12], [6], and in these cases the averaging timescale and the overall time of measurement wildly affect the measure. In fact, in this case the capacity of the wireless link is time-varying, and depends also on the packet size and the cross-traffic intensity (due to channel contention) [4], [8]. Moreover overheads of communication initiation, neighbor nodes interference, and time variability of SNR (Signal to Noise Ratio) all make capacity (and ABw) specifically hard to estimate in wireless and mixed wired-cum-wireless scenarios. As a consequence, ABw estimation is not a trivial task, and many tools and techniques have been proposed, often with specific intended application scenario (and associated limitations). We refer to [9], [4] for a comparison and analysis of different ABw estimation tools on wired and wireless paths.

B. Software-Defined Networking

Software-Defined Networking (SDN) is a recent common name for earlier approaches in network devices management, rooted in works in the fields of active networks and network virtualization. In the current form, SDN attracted significant research and deployment after the introduction of OpenFlow [13] standard protocol for controller-switch communication.

In this paper we adopt the definition of SDN given in [11], based on the following four elements: (i) control and data planes are separated from each other: network devices are simple forwarding elements *SDN switches* or simply *switches*; (ii) forwarding rules are based on a set of fields in the packet headers; this enforces unified behavior of networking elements regardless the high-level function and the protocol layer they act on (switches, routers, firewalls, etc.); (iii) control plane is moved to an external entity called the *Network Operating System* (NOS) or SDN controller. For more details and a broad discussion on SDN we refer to [11]. We have analyzed ABw estimation with *passive* measurement methods in [14], to which we refer for an in-depth analysis. In this paper we aimed at estimating the performance effects from the introduction of an SDN switch in the process of *active* network measurement, in exchange for the flexibility and the generalization provided by SDN management.

C. MONROE testbed

From the point of view of the experimenter, the platform is composed of the following main components: (i) geographically distributed hardware appliances (MONROE nodes) running the experiment software; (ii) the software running on the MONROE node (divided in core components and user-defined experiments); (iii) the management system, allowing user access, experiment scheduling, and data import; (iv) a database holding experiments data and automatic periodic measurements.

The hardware setup is standardized on all nodes. The software defining the experiments is executed on the MONROE node in light-weight virtualized environments (Docker containers). For further details on MONROE we refer to [5], [1] and cited deliverables.

III. SOMETIME RESEARCH PLAN

According to the increasing interest in the MBB performance, a number of mobile testbeds has been deployed during the last years. Many of them share common constraints in terms of HW resources, further exacerbated by the virtualization technologies often leveraged today. Moreover, additional concerns are generated by data plans made available by mobile operators, which may represent a severe limitation to the measurement traffic possibly generated by the tools adopted by MBB platform users.

In this framework, with reference to the aforementioned MONROE project, the main intent of SOMETIME is to provide experimenters with a highly valuable tool to measure the ABw in MBB scenarios. The project aims at providing the estimation of ABw by active measurements leveraging the SDN paradigm, both to tune the technique considering interference with node-local processes (that is a more realistic scenario compared with mutually exclusive measurements), and to mitigate such interference. Possible applications for experiments in MONROE (e.g., multimedia streaming to smartphones, tablets, in-vehicle-infotainment systems) present the additional challenge of possible sharing of computing and communication resources during the measurements: to emulate, assess and mitigate this interference we will adopt an SDN-based approach to manage traffic in the mobile terminal (MONROE node). This approach will have the additional outcome of evaluating the possibility of running multiple active experiments concurrently on the MONROE node, guaranteeing isolation of traffic engineering or routing set-ups required by the different experimenters.

To the best of our knowledge, our proposal to use an SDN controller to isolate measurement traffic from other network traffic generated or received by the terminal equipment in a mobile broadband scenario has not been considered yet in literature, thus its design, implementation, and experimental evaluation in MONROE will constitute another highly valuable contribute to the current state-of-art. It is worth to notice that, although our experimentations are tailored on the MONROE platform, the outcomes carried by the SOMETIME analyses are of general interest, also due to the common characteristics and the typical issues related to MBB platforms.

In the following the main steps planned for SOMETIME are briefly described.

(a) Evaluation of the suitability of publicly released ABw estimation tools for the considered scenario (MBB test platform).
(b) Evaluation of the impact of the HW and of the (lightweight) virtualization layer on the accuracy of traffic generation tools.
(c) Evaluation of the impact of SDN technologies on the accuracy of traffic generation tools.
(d) Definition, setting, and evaluation of an SDN-enabled ABw estimation tool tailored for the MONROE measurement scenario.
(e) Leveraging the SDN-enabled setup, adding ABw estimation to the set of metrics collected by MONROE, and characterizing the MBB networks considered in MONROE in terms of this metric.

Step (a) is necessary as literature on ABw estimation tools has found their performance to significantly depend on the measurement context [4]. Step (b) is suggested by the limited-resource nature of both the measurement platform MONROE and of the real-world scenario it models, and is necessary to properly tune the active-measurement processes. Following the previous one, step (c) will allow to evaluate the trade-off implied by the adoption of SDN: on the one hand, SDN is introduced to assess and mitigate the interference of local traffic and the measurement traffic, also providing a standard control interface, on the other hand, its presence will put additional burden on the limited-resource measurement environment, possibly interfering with both the local traffic and the measurement tools. The analyses above can be performed according to several metrics of interest (e.g., achievable throughput, inter-packet-time generation/reception precision, etc.). The choice of the metric may depend upon on either the specific phase of the project or the granularity of the information the experimenters are interested in. Finally, step (e) will conclude and fulfill the ultimate goal of the research project.

In this paper, we present a set of preliminary analyses covering steps (a), (b), and (c). In more details, we first provide an experimental comparison of the the ABw estimation tools (Section IV) on HW modeling a MONROE node. We then analyze the packet generation accuracy in MONROE setup through virtualization (Section V). Then, we assess and compare the packet generation limits with and without SDN, on bare HW and in virtualized environment (Section V). Finally, we collect and discuss experimental measurements of UDP achievable throughput from the actual MONROE platform (Section VI).

IV. Comparing ABw estimation tools in SDN

Based on the big picture offered by the related literature, we have chosen a subset of tools in order to perform the estimation of the ABw in our SDN-based scenario. The criteria used to select the candidate tools are: (i) the availability of the source code and the possibility to correctly compile it for Debian *jessie* (same as deployed on MONROE and latest stable release); (ii) the enhancement technique adopted by each tool to improve accuracy and to mitigate intrusiveness (aiming at extending the variability of techniques tested in our environment).

These criteria have led us to choose: (i) *Pathload*, because it has proven to be the reference for accuracy, being also stable with the default parameter values over the many different scenarios in which it has been compared with other tools [4]; (ii) *YAZ* [16], that has been specifically designed to improve on Pathload by reducing its intrusiveness; (iii) *ASSOLO* [9], because it has been implemented with support for real-time kernel facilities; (iv) *STAB* [15], because it implements per-hop ABw estimation, with detection of the thin link (i.e. the link introducing a reduction of the ABw of the path: the last thin link is the tight link).

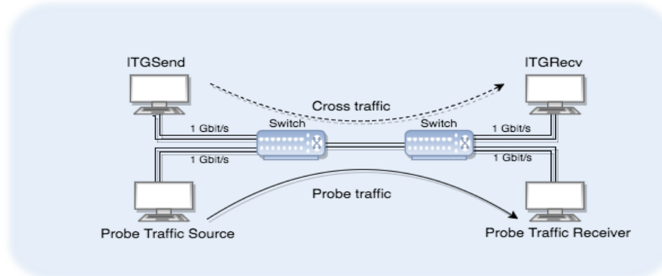


Fig. 1: Setup of SDN testbed for ABw estimation.

We performed an experimental evaluation of available tools for ABw estimation, using an emulated SDN testbed based on Mininet [10] and adopting the most popular software implementation of an SDN switch: Open vSwitch (OVS) [2]. The setup of the testbed is shown in Figure 1, modeling a MONROE node hosting the ABw estimation tool (Probe Traffic Source) that performs a measurement of the network path that leads to the measurement server, hosting the ABw receiver component (Probe Traffic Receiver). The link between MONROE node and measurement server is terminated by two OVSes and is shared with concurrent cross traffic, that in the testbed is generated by ITGSend, towards ITGRecv.

To evaluate the performance of the tools we run 10 repetitions of each experiment in the same conditions, with no appreciable qualitative variation in the results. In the following we report a summary of results over 44 different combinations of capacity and cross-traffic for the tight link (that was also the narrow link), with either Poisson or Constant Bit Rate (CBR) cross-traffic. As outcomes with Poisson cross-traffic resulted equivalent to CBR related ones, only the former have been reported.

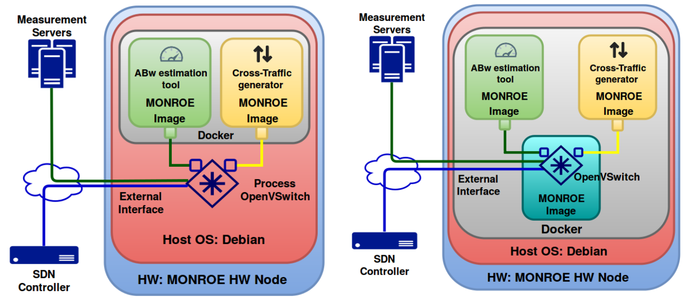
We have defined as *convergence* metric the percentage of experiments in which the tool produced a result (in some cases, as per Yaz and ASSOLO, the tool frequently failed to return a result, or hanged). We noticed that Pathload—one of the first tools to be implemented and distributed—performs best for both convergence and accuracy.

More detailed data about accuracy is reported in Table I, where *Capacity* (C) is the network capacity of the narrow link; *Cross traffic* (Xt) is the bit rate of traffic that traverses the narrow link (making it the tight link, too); *Outcome* (O) is the average of the values returned by the tool execution (for successful runs only); and *Relative Error* (RE) is $RE = \frac{O-(C-Xt)}{C-Xt}$.

Combinations of Capacity and Cross-traffic that are not shown have led to hanging of tool, or termination with lack of a response. It can be noted that even in the best cases (Pathload and STAB) the average relative error never goes below 21% of the expected result. These outcomes suggest to perform

TABLE I: Accuracy of considered tools.

Tool	Capacity (Mb/s)	Cross-traffic (Mb/s)	Result (Mb/s)	Relative Error
Pathload	3	1.5	2	0.33
	5	1.5	2.5	-0.28
	100	74	20	-0.23
Yaz	10	4	2.4	-0.60
ASSOLO	10	2	34	3.25
	20	6	34	1.42
	100	76	48	1.00
STAB	5	1.5	3.7	0.5
	10	4	4.7	-0.21
	100	74	32.6	0.25
	1000	74	118	-0.87



(a) *Host-OVS* setup. (b) *Docker-OVS* setup.

Fig. 2: Considered testbed configurations.

further investigation to find a suitable method for active ABw estimation in the MONROE testbed.

As accuracy in packet generation heavily affects accuracy of active ABw estimation tools, specially for high-speed networks, in the following section we discuss the impact of SDN and lightweight virtualization on traffic generation accuracy.

V. Impact of SDN and lightweight virtualization on Packet generation

In the process of active ABw estimation, the generation of probe traffic is performed at given bit rates, or equivalently specifying given inter-packet times. Motivated by the experimental results described in the previous section, we have investigated the impact of SDN and lightweight virtualization on packet generation accuracy. For this reason we have setup a node in which multiple flows are generated and traverse an SDN switch, from which they are forwarded through the mobile interface to the mobile access network, occupying concurrently the wireless link (that is the most likely to be the narrow link in paths towards online services).

A. Experiments setup

Three different configurations have been considered for the evaluation of the impact of virtualization and SDN environments on ABw estimation (specifically for the traffic generation performance) as follows.

Native. The application that generates/receives probe traffic runs in the host operating system, with no virtualization and no SDN switch forwarding the traffic. This setup is considered as a benchmarking reference for what concerns the MONROE node, while is the preferred implementation for the measurement server counterpart.

Host-OVS. The application that generates/receives probe traffic runs in a Docker container, and the probe traffic is forwarded through an SDN switch (OVS) that is run in the host operating system (Figure 2a). The SDN switch manages all network traffic interesting the node. To be implemented in the MONROE testbed this setup requires the extension of current testbed with an OVS service running in the host OS of the node, and configuration entry points for the network (e.g. in the form of OpenFlow messages, or leveraging an SDN Controller Northbound API).

Docker-OVS. In this configuration both the application that generates/receives probe traffic, and the software SDN switch, run in Docker containers (Figure 2b). This setup is the closer to the current deployed implementation of the MONROE testbed. In order to implement it the testbed should be extended with the possibility of deploying multiple containers in a single experiment, and specific capabilities should be allowed for these containers.

For the experimental evaluation we have chosen the RYU SDN controller [3], because of its lightweight nature and the ease of its programming. In our experimental setup the SDN switch is configured to act as a simple *learning switch*, with the standard FIFO scheduling policy. These choices do not affect the generalizability of the experimentation.

B. Experimental results and discussion

To analyze how the process aiming at measuring the achievable throughput is biased by the virtualization implemented by the Docker environment, we setup the experimental testbed according to the three configurations above. The overall testbed has been configured on two physical machines, connected back-to-back through a CAT5E Ethernet cable. The tools configured according to the three configurations above have been installed onto the sender node. The receiver node is an external machine (in a native configuration) thus acting as a measurement server as designed for the final architecture.

In order to measure the achievable throughput, we saturated the link generating UDP traffic using D-ITG [7]. We observed a notable discrepancy between the required bit rate and inter-packet time, and those of the traffic actually generated. In addition, when requiring D-ITG to generate traffic at a given bit rate (that is achievable, as actually observed at the receiver side when requiring a higher bit rate at sender side) the observed bit rate anomalously further decreases. Therefore we set the sending rate at a value as higher as possible.

Figure 3 shows the results of an experimental campaign made up of 100 repetitions of the experiment described above. As shown in the figure, the best performance is obtained when running the experiment in a native environment, as expected. When OpenVSwitch is placed onto the host OS, the monitored performance is still acceptable. When the Docker-OVS configuration is run, we obtained the worst performance. In this last scenario—also designed according to the constraints of the MONROE project—the host network is connected to the docker container through a bridge that also implements NAT functions, and potentially is the source of further performance limitation.

Another set of experiments we run required two distinct containers to generate two (either CBR or Poisson) flows. Two distinct processes in place of containers are considered in the case of the native environment configuration. During

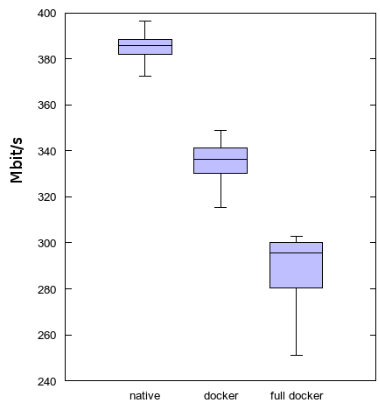


Fig. 3: Comparison of upstream achievable throughput. The boxplots report maximum, 75th percentile, median, 25th percentile, and minimum.

TABLE II: Configuration and measurement results for CBR and Poisson flow generation. Packet size was set to 1470B, flow duration was set to 10s. Shown values are averages over 100 repetitions.

(a) CBR.

		Configuration		
		Native	*Host-OVS*	*Docker-OVS*
Packet rate (pkts/s)	requested	75080	70080	50080
	obtained (Flow1)	5143	5529	4814
	obtained (Flow2)	4986	5282	5027
Bit rate (Mb/s)	requested	882.9	824.1	588.9
	obtained (Flow1)	60.5	65.0	56.6
	obtained (Flow2)	58.6	62.1	59.1

(b) Poisson.

		Configuration		
		Native	*Host-OVS*	*Docker-OVS*
Packet rate (pkts/s)	requested	75080	70080	50080
	obtained (Flow1)	5538	5235	5450
	obtained (Flow2)	5630	5662	4635
Bit rate (Mb/s)	requested	882.9	824.1	588.9
	obtained (Flow1)	65.12	61.57	64.09
	obtained (Flow2)	66.21	66.58	54.51

this experimentation, we noticed that the link saturation—forced by the generated synthetic traffic—caused the D-ITG signaling traffic to be partially dropped and therefore subjected to retransmission. These retransmissions heavily biased the results of the measurement process. Although we could have mitigated this issue by configuring the environment to use a dedicated interface for transmitting signaling traffic, we decided not to implement this solution, as it cannot be ported to the final MONROE setup. Therefore we set the requested bit rate to lower values, thus adopting a more conservative position. Traffic generation has been repeated 100 times.

Table IIa reports the results in the case of two CBR flows generated for the three cases described above. We can notice how flows fairly share the link whereas their variability is lower in the case of Docker-OVS configuration. The median value is comparable for all the cases. Finally, when considering two Poisson flows (see Table IIb), higher variability has been observed. Stabler results have been obtained only for the Docker-OVS configuration.

The challenging nature of ABw estimation in SDN scenarios becomes evident, from both the evaluation of the tools and the assessment of configuration overhead on traffic generation performance. The results showed high variability, and sensitiveness to virtualization and configuration overhead introduced by the additional presence of an SDN switch on the network path to the measurement server. Traffic generation performance is therefore the first aspect to be investigated in order to inform tool selection and configuration. This lead us to perform a preliminary campaign of MONROE test nodes to assess the traffic generation capability,as described in the next section.

VI. Preliminary field experiments

Leveraging the experience acquired thanks to the experimentations described above, we designed an experimental campaign to be deployed and run onto the MONROE testbed. The goal of this experimental campaign is to assess the possible limitations of MONROE nodes primarily in terms of traffic generation capability, and thus in performing network throughput/available bandwidth measurements. This campaign also constitutes a first hands-on exploration of MONROE services and tools in the course of the SOMETIME project.

A. Experiments setup

The experimental setup is composed of (i) a small number of MONROE nodes playing the sender role and (ii) a measurement server deployed at the University of Napoli (with 1 Gbps network interface). Onto these MONROE nodes the D-ITG sender component (ITGSend) was run. These nodes are required to generate UDP traffic at a packet rate as high as possible towards the receiver component (ITGRecv) running on the measurement server. More specifically, each node is required to perform the experiment in two phases: the payload size is equal to 1 byte for the first phase and to 1450 bytes for the second, while all the other parameters are kept fixed. It is worth noting that the first case (lower dimension of a packet with payload) is intended to stress CPU usage, for full network stack execution only, while the second (upper bound for packet size such that fragmentation is avoided) adds a toll on bandwidth resources (and thus buffers and I/O operations). Therefore this experiment allows to evaluate the overall range of traffic generation capability (in terms of bit rate and packet rate) of the node, as it could be required by an ABw estimation tool (hence the requirement of absence of fragmentation and the use of the UDP protocol).

For this experimentation 4 MONROE testing nodes placed in 4 different countries (Norway, Sweden, Spain, and Italy) have been selected.

B. Experimental results and discussion

The results of the experimental campaign described above are reported in Table III, where the results at sender side are shown. Note that these values are only related to the generation capability of the sender and they primarily depend upon the MONROE nodes and the performance of their implementation.

In general we experimented low variability in our results, revealing that no major discrepancy exists among the performance of the nodes we tested. For the first phase (1-byte payload) we found that packet rate is 18939.33 ± 719.3 pkts/s,

TABLE III: Sender-side results for MONROE Testing nodes.

Node ID	Country	Operator	payload:1B bit rate (Mb/s)	payload:1B pkt rate (Kpkts/s)	payload:1450B bit rate (Mb/s)	payload:1450B pkt rate (Kpkts/s)
201	Norway	Telenor	3.13	19.56	137.51	11.85
248	Sweden	TelenorS	2.95	18.43	122.33	10.55
58	Spain	voda ES	2.89	18.04	133.61	11.52
119	Italy	I WIND	3.16	19.73	122.68	10.58

on average, while for the second phase (1450-byte payload) the observed packet rate is 11123.66 ± 575.13 pkts/s. As expected a higher packet rate is observed when issuing packets with 1-byte payloads. This value represents the maximum number of packets per second D-ITG is able to issue when running on MONROE nodes: it is not limited by the network bandwidth available, being dependent from the computational capability of the nodes and/or by performance of the virtualized stack implementation.

As reported in the table, we obtained bit rates ranging from 2887 to 3156 Kb/s, and from 122334 to 137508 Kb/s for the 1st and the 2nd phase, respectively. This discrepancy among the two phases is due to the protocol overhead (2800% and 1.9% for the first and the second phase, respectively). The bit rate measured with packets having 1450-byte payload (equal to 129034 Kb/s, on average) is the maximum rate at which a MONROE node is able to inject data into the network (note that is has to be intended as a goodput value). This value, being higher than throughput values measured at receiver side (21674 Kb/s, at most), indicates that the generation capabilities of our experimental setup (i.e., D-ITG over MONROE nodes) is in line with the requirements of the experimentation and therefore is suitable to be adopted in the analyses that will be performed within the SOMETIME project. Ongoing investigation and experimenting regards the implementation in MONROE of the Docker-OVS setup including Open vSwitch, in order to assess the overhead of the presence of the SDN on the measurement path.

VII. Conclusion

To evaluate and experiment with state-of-art ABw estimation tools in mobile broadband access networks we tested several ABw tools in an emulated SDN-based testbed, under controlled conditions. The experimental evaluation has shown that all considered tools offer poor accuracy and medium/poor reliability (often not converging to an estimate, because they either failed to return a measurement result, or hanged until forced termination). Due to the poor accuracy even of the best tools, we have conducted a deeper analysis of the traffic generation aspect. To this aim, we designed an SDN testbed reproducing different alternate configurations for implementing ABw estimation in SDN environment on MONROE nodes. To assess the impact of setup overhead on traffic generation performance on actual MONROE nodes we deployed an experiment on Testing nodes, using the D-ITG traffic generator to stress the generation capabilities of the node. From the analysis of the results we verified that D-ITG is able to generate enough traffic to fully saturate the upstream link, thus current choice of MONROE HW does not prevent per-se the accurate estimation of ABw with active methods. We are conducting further experiments to estimate the additional impact of the OVS switch.

As future work, we plan to experimentally evaluate other possible setups (such as running OVS in a full-virtualized environmemt, managing containers communications); moreover we will explore SDN programmability to both perform traffic monitoring (for real-time adjustment of ABw estimation) and traffic engineering (evaluating isolation of probe traffic and cross traffic).

Acknowledgment

The research described in this paper has been partially funded by the SOMETIME project of the 1^{st} MONROE Open Call, funded from the European Union in the Horizon 2020 research and innovation programme under grant agreement No 644399 and art. 11 DM 593/2000 for NM2 srl.

References

[1] The monroe project website. https://www.monroe-project.eu.

[2] Open virtual switch (open vSwitch).

[3] Ryu controller. https://osrg.github.io/ryu/.

[4] G. Aceto, A. Botta, A. Pescapé, and M. D'Arienzo. Unified architecture for network measurement: The case of available bandwidth. *Journal of Network and Computer Applications*, 35(5):1402 – 1414, 2012. Service Delivery Management in Broadband Networks.

[5] Ö. Alay, A. Lutu, R. García, M. Peón-Quirós, V. Mancuso, and et al. Measuring and assessing mobile broadband networks with monroe. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, June 2016.

[6] L. Angrisani, A. Botta, A. Pescape, and M. Vadursi. Measuring wireless links capacity. In *Wireless Pervasive Computing, 2006 1st International Symposium on*, pages 5–pp. IEEE, 2006.

[7] A. Botta, A. Dainotti, and A. Pescapè. A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*, 56(15):3531–3547, 2012.

[8] F. Chen, H. Zhai, and Y. Fang. Available bandwidth in multirate and multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 28(3):299–307, April 2010.

[9] E. Goldoni and M. Schivi. *End-to-End Available Bandwidth Estimation Tools, An Experimental Comparison*, pages 171–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[10] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pages 253–264, New York, NY, USA, 2012. ACM.

[11] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, Jan 2015.

[12] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 314–321, New York, NY, USA, 2004. ACM.

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Computer Communnication Review*, 38(2):69–74, Mar. 2008.

[14] P. Megyesi, A. Botta, G. Aceto, A. Pescapè, and S. Molnár. Available Bandwidth measurement in Software Defined Networks. In *Proc. of the ACM/SIGAPP Symposium on Applied Computing*, Pisa, Italy, Apr 2016. ACM.

[15] V. J. Ribeiro, R. H. Riedi, and R. G. Baraniuk. Spatio-temporal available bandwidth estimation with stab. In *ACM SIGMETRICS Performance Evaluation Review*, volume 32, pages 394–395. ACM, 2004.

[16] J. Sommers, P. Barford, and W. Willinger. Laboratory-based calibration of available bandwidth estimation tools. *Microprocessors and Microsystems*, 31(4):222 – 235, 2007. Special Issue with selected papers from the 11th {IEEE} Symposium on Computers and Communications (ISCC'06).