Making Trustable Satellite Experiments: an Application to a VoIP Scenario

Antoine Auger
TéSA laboratory
antoine.auger@tesa.prd.fr

Emmanuel Lochin
ISAE-SUPAERO
University of Toulouse
emmanuel.lochin@isae-supaero.fr

Nicolas Kuhn
CNES
nicolas.kuhn@cnes.fr

***Abstract*—How many times have ever asked yourself: "Can I trust my satellite experiments' outcome?". Performing experiments on real satellite system can either be (1) costly, as the radio resource may be scarce or (2) not possible, as you can hardly change the waveforms transmitted by the satellite platform. Moreover, assessing user applications QoE can hardly be done using only simulated environments while the QoS modeling of a satellite system can often lead to non-conclusive or ambiguous results. The aim of this paper is to bring out representative solutions allowing the networking community to drive consistent experiments using open-source tools. To this end, we compare Mininet and OpenSAND satellite emulator to a real satellite access provided by CNES. We consider VoIP traffic to analyze the trade-off between reliability of the results, ease of use and reproducibility of the experiments.**

I. Introduction

Following the Satellite Industry Association (SIA), the global revenue for ground equipment moved from $12.8 billion to $18.5 billion considering only consumer equipments and home services such as satellite VoIP, broadband and mobile. This trend clearly illustrates the growth of the satellite market[1]. This trend can be explained by the fact that satellite communications provide global coverage, allowing to connect isolated areas (e.g., low population density regions, deserts, offshore oil rigs), to reduce digital exclusion[2] or even to handle mobility aspects (fleets, maritime transport, Argos beacons, etc.).

Furthermore, one of the most challenging satellite market today deals with flight communications. This trend is going to increase over the next years as several flight companies now provide on-board Internet services (FlyNet, Nordic Sky, etc.). These services are generally advertised as a transparent extension of the standard Internet allowing to enjoy your own everyday applications on-board. This suggests to the users that they could expect the same service than a terrestrial access. However, as the use of your own Internet applications over long-delay links might lead to discrepancies, satellite vendors first need to understand the characteristics of each application in order to optimize their satellite access. Secondly, they need to assess the experience perceived by the user (Quality of Experience - QoE) instead of standard networking metrics (Quality of Service - QoS).

We also observe further interest for this communication scheme, as for instance illustrated by the IETF working groups exploring the use of novel transport protocol mechanisms (e.g., Google QUIC for satellite) and applications (network coding for satellite [1]).

All these arguments motivate the present study where one of the main problem is to choose the right tools and metrics in order to evaluate the QoE of user applications and to improve their performances. In this paper, we propose an analysis followed by recommendations and guidelines on how to perform trustable satellite experiments. Considering a VoIP scenario, we designed several experiments that will be played over various satellite measurement systems:

- traffic is generated over a real satellite link, which has been provided by CNES[3] towards CESARS platform [2];
- outcome metrics are compared to those obtained with:
  - Mininet [3], a well-adopted network emulator that has gained popularity for its capability to reproduce trustable experiments;
  - OpenSAND [4] satellite emulator.

For each of the three proposed evaluation tools, we analyze the trade-off between reliability of the results, simplicity to use and reproducibility of the experiments.

Our contribution is twofold. First, we present a VoIP scenario where we compare QoE for two codecs (G.711.1 and G.723.1) in a geostationary satellite system (GEO satellite). Second, we use these QoE results to give recommendations and guidelines on satellite measurement systems according to the metrics of interest.

II. Required Background and Related Work

Before presenting the evaluations performed, we wish to introduce important metrics used in the following and justify some choices to exclude simulation software.

A. Voice over IP and QoE

The main difference between QoS and QoE is that the former relates to networking metrics (delay, jitter, etc.) while the latter deals with subjective metrics linked to user experience (VoIP or video perceived quality). QoE metrics are

[1]See https://www.sia.org/wp-content/uploads/2017/07/SIA-SSIR-2017.pdf
[2]See https://irtf.org/gaia
[3]The Centre National d'Etudes Spatiales (CNES) is the French national space agency

specific to a class of applications or services. We lay on this paper on those standardized by the ITU-T as they have been adapted for IP communications.

In the following, we specifically focus on the VoIP service. According to ITU-T's recommendations [5], QoE characterization for VoIP generally consists in computing the Mean Opinion Score (MOS) from the R-factor metric. Several parameters (VoIP codec used, etc.) and QoS metrics (one-way delay, packet probability loss, etc.) are used as inputs to compute these resulting QoE metrics as in [6]–[8].

B. Approaches for Satellite Experiments

As for all networking systems, three main approaches exist to perform satellite experiments, namely simulation, emulation and real-field experiments [9], [10]. Here, we focus on the evaluation of emulation against real-field experiments. Although simulation offers a convenient way to analyze networking behaviors, it does not provide an easy way to assess the performance of real applications. This makes emulation to derive metrics related to QoE while simulation mainly focus on QoS results.

We point out that ns-3[4] now provides *Direct Code Execution* to test your own applications. Nevertheless, this feature requires to modify the application source code to import mandatory ns-3 librairies [11]. As a result, this limitation excludes the ability to evaluate proprietary Internet applications, justifying the fact that we did not consider this approach.

III. Practical QoE Assessment for Satellite VoIP

We propose to perform QoE assessment for VoIP over a GEO satellite access also called GEO satellite VoIP. Such scenario is a good illustration of the use cases presented in the introduction such as emergency or maritime communications. Furthermore, satellites are particularly relevant to support VoIP for three main reasons. First, they provide more reliable, harder to disrupt and safer connections than terrestrial technologies. Then, three GEO satellites can almost provide global coverage, even for sparsely populated areas. Finally, VoIP codecs enable low-bit-rate communications (of about a dozen kbit/s), allowing a GEO satellite to provide multiple "side services" while maximizing the capacity usage.

A. Hypothesis

Starting with a real satellite access [2] as a "reference setup", we compare QoE results to those obtained with Mininet [3] and OpenSAND [4], two well-adopted network emulators, for the same VoIP experiments. Our goal is to give orders of magnitude of QoE for satellite VoIP, evaluate pros and cons of each setup before analyzing fidelity and giving some recommendations regarding use cases for satellite VoIP.

We used the D-ITG traffic generator [12] to generate VoIP traffic (i.e., flows of UDP datagrams) from satellite terminals to satellite gateways, emulating one-way VoIP calls on the return link. The return link has been chosen given its smaller capacity compared to the forward link, allowing us to better visualize QoE variations due to packet losses under high VoIP traffic. To better understand the influence that a codec may have on QoE, we selected the G.711.1 and G.723.1 codecs that have constant but different bitrates of 84.8 kbit/s and 15.184 kbit/s, respectively (total bitrates for Ethernet frames with the D-ITG *Voice Activity Detection - VAD* option enabled).

[4]Official website: https://www.nsnam.org/

B. Experimental Procedure

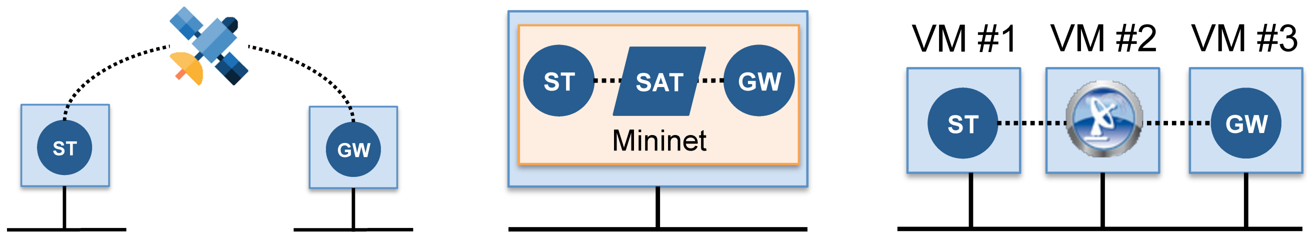
We call "VoIP experiment" the process of computing QoE metrics (R-factor and MOS) for a specific setup (real satellite, Mininet, OpenSAND) where a given codec (G.711.1, G.723.1) is used to send n concurrent one-way VoIP flows. We vary the number of VoIP flows to emulate network load in order to progressively saturate satellite link. Each VoIP call lasts 60 s. Finally, we deployed three experimental setups used to run different VoIP experiments (see Fig. 1). Given the time allotted for our real-field experiments and given the excellent link stability for the GEO satellite, we decided to only run each experiment 3 times in order to compute mean value and standard deviation error.

The first setup corresponds to a real satellite access (see Fig. 1a) and represents our reference scenario for further comparisons. Real-field tests took place on June, 13-14 2018 on the CNES' site in Toulouse. The return link of the GEO satellite used as part of the CESARS platform has the following characteristics: 2.604 MBd symbol rate, QPSK $\frac{2}{3}$ modulation, 3 Mbit/s capacity. During these two days, we observed clear sky conditions and, as a result, a very stable link for the GEO satellite used. It should be noted that we were able to use a complete satellite access (with both Satellite Terminal - ST and Gateway - GW). Preliminary sanity check tests allowed us to characterize more precisely the satellite link, indicating a capacity of roughly 3 Mbit/s for the return link.

In addition to this reference setup, we deployed one Mininet and one Opensand setup (see Fig. 1b and Fig. 1c, respectively). For both setups, we aimed to mimic as much as possible the behavior of the real GEO system. To this end, we manually tuned several configuration parameters to best meet the real satellite configuration presented above. For Mininet, we defined two hosts (*ST* and *GW*) connected through a switch called *SAT*. Both links *ST-SAT* and *SAT-GW* have a capacity of 3.5 Mbit/s, a constant delay of 125 ms, 0% of loss probability while having an associated queue size of 1 000 packets maximum. Our second setup consists in an OpenSAND deployment on three virtual machines running the different components of OpenSAND with sufficient resources and custom configuration.

C. Results and Analysis

As previously mentioned, two QoS metrics provided by the D-ITG software were of interest for our study. Thus, percentage of packets dropped has been used to compute the *"random packet-loss probability"* while the maximum one-way delay has been used in place of *"mean one-way delay of the echo path"* and *"absolute delay"* parameters according to



(a) Real satellite access (CESARS) (b) Mininet standalone (c) OpenSAND emulator

Fig. 1: Overview of the three experimental setups selected for comparison. VoIP flows are sent from the Satellite Terminal (ST) to the Gateway (GW). VM: Virtual Machine.

ITU-T's nomenclature. We point out that we cannot compare QoS metrics as they are obtained over different systems. This means that such comparison can only be done considering the trend of the results and not the values obtained.

From the MOS results (see Fig. 2a and Fig. 3a), we can identify two different states:

1) **Initial**: when there is only one VoIP flow;
2) **Concurrent**: when there are several VoIP flows, leading to a saturation point we estimate below.

Please note that the saturation point varies from one codec to another. For instance, with the G.711.1 codec, the saturation point corresponds to approximately 35 flows ($3 * 10^6/84.4 * 10^3$) concurrently transmitted on the return link. For the G.723.1 codec, the saturation point is reached starting from 197 flows.

Considering the real satellite experiments as reference results, we notice that OpenSAND gives a MOS of 3.8 and 3 in initial state for G.711.1 and G.723.1, respectively. In concurrent state, however, Mininet results are closer to satellite results. Above 40 flows, the link is saturated and we observe that OpenSAND gives better results than Mininet (see Fig. 2a). As expected, due to the low bitrate of G.723.1 codec, saturation point is not visible on Fig. 3a. Actually, MOS values only return subjective metrics of the effective quality perceived by end-users when performing VoIP calls over satellite links.

To conclude with these MOS results, we observe an important trend which highlights the impact of the emulation system over the results. Below the saturation point, the reader might notice that satellite results are best approximated by Mininet than OpenSAND while the latter gets better approximations above the saturation point. We recall that the MOS computation is not linear, giving only a trend that does not allow us to directly conclude without conjointly considering QoS metrics. We give further explanations on how such results should be carefully interpreted in Section IV-B.

We now turn to the two QoS metrics used to compute the MOS, namely the maximum one-way delay and the percentage of packets dropped. The percentage of packets dropped allows us to estimate the saturation point. For G.711.1 codec, we clearly see on figure 2b that losses start to occur from 40 flows, which is consistent with our previous estimation (35 flows). For G.723.1 codec, we notice few losses for the satellite setup that can be neglected (see Fig. 3b). As previously said, packet losses only explain in part the MOS variations.

Fig. 2c shows an increase of the delay for G.711.1 codec above the saturation point. Thus, from 30 to 40 flows, delay increases of 150% and 100% for both satellite and OpenSAND setups, respectively. Several investigations and some basic notions in queuing theory showed that this increase could be explained by excess buffering in intermediary routers, queues and other application buffers. This phenomenon is also well-known as *bufferbloat* due to the fact that we are in steady-saturated regime. In contrast, it is interesting to notice that Mininet results do not show any bufferbloat issue. This shows that, above the saturation point, there is a real interest for using a satellite emulator instead of a simple network emulator. While OpenSAND configuration can be more complex to tune, we clearly see that its numerous parameters allow to gain in terms of fidelity. For G.723.1 codec, the situation is slightly different as we do not reach the saturation point. However, OpenSAND results clearly show a maximum delay that increases from 0.3 s for 1 flow to 1.2 s for 40 flows.

We attempt to discuss these observations in the following.

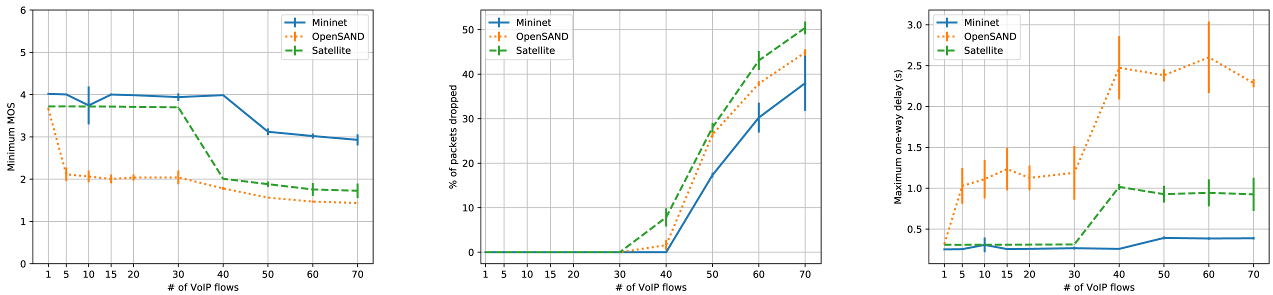
IV. On the Trustability of Satellite Experiments

As we previously observed, making trustable satellite experiments is a challenging task. This section aims to weight up the impact of emulation systems in the sake of evaluating QoE over SATCOM. We do not pretend to explain all the issues observed but attempt to provide a comprehensive summary that we also present in Table I.

A. Comparison of Approaches

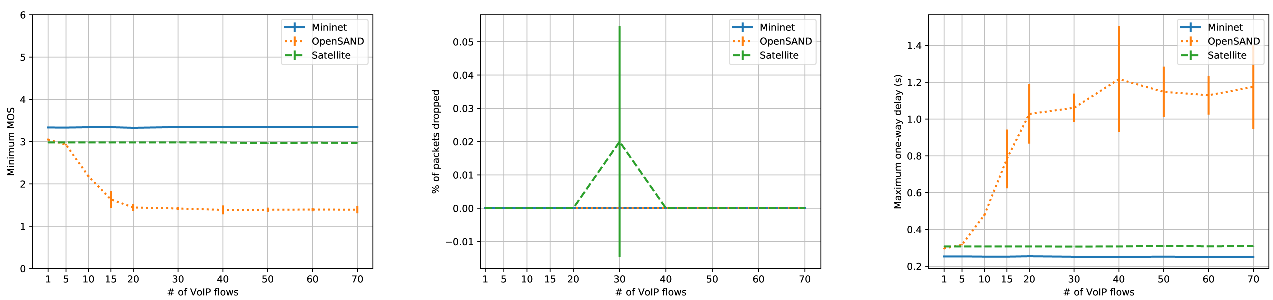
As for all emulator systems, both Mininet and OpenSAND rely on hypothesis and simplifications in order to approach reality as much as possible. While complexity does not necessarily represent a guarantee in terms of result fidelity, specialized satellite emulators may better model some satellite features like resource allocation, frame (dis)encapsulation, attenuation, etc. For instance, we saw that a satellite emulator like OpenSAND provides better MOS estimation than Mininet above the saturation point.

Of course, a real satellite access offers unrivaled realism while allowing to benefit from advanced satellite features like Performance-Enhancing Proxies (PEPs). However, without a



(a) MOS (b) % of packets dropped (c) Max. one-way delay

Fig. 2: Experimental results for G.711.1 codec



(a) MOS (b) % of packets dropped (c) Max. one-way delay

Fig. 3: Experimental results for G.723.1 codec

partnership with a space agency, accessing to a satellite infrastructure may be difficult.

Mininet is an interesting solution, easy to use, requiring at least one single machine to virtualize network plane. Since this software is not a satellite network emulator, users can only create simple satellite topologies by using Mininet's API. Limitations occur when users require more advanced features like resource allocation, modulation changes, etc.

On the contrary, OpenSAND is a mature software that can be used to emulate a complete end-to-end satellite communication system. Numerous parameters can be tuned like carriers, modulation types, etc. As a counterpart, OpenSAND installation should be done on three hosts, which complexifies system management. Note that, to speed up experiment deployment, CNES also provides OpenBACH[5], which is a modular software for benchmarking and metrology for IP networks in general.

B. Result Fidelity and Use Cases

In the following, we use the notation *Satellite*, *OpenSAND* and *Mininet* to refer to the corresponding curves in Figures 2 and 3, and OpenSAND and Mininet when we refer to the software.

With a simple VoIP scenario, we showed that emulation could be used to highlight trends, close enough to a real GEO system. We show that Mininet and OpenSAND provide good MOS approximations if the experimenter is aware of the limitations previously highlighted (below or above the saturation point for instance). In particular, both OpenSAND and Mininet manage to correctly model link satellite capacity.

However, our VoIP experiments showed that Mininet failed to give correct results regarding average jitter (results not provided here as not used in MOS computation). One important lesson learned is that the complexity of satellite systems (with intermediary queues, encapsulation, etc.) cannot be reduced to two hosts connected by a high-latency link. On this point, OpenSAND provides an emulation of a complete end-to-end satellite system.

Comparing QoS metrics from different setups must always be done by considering the trend and not only the values themselves. Taking as an example Fig. 2c, although the distance between *Mininet* and *Satellite* is smaller than with *OpenSAND*, the trend is not. In this figure, we observe both the slopes of *OpenSAND* and *Satellite* clearly evolve in the same way.

We observe that OpenSAND applies a small but constant overhead of 1.2 s due to packet processing, even below the saturation point as shown in Fig. 3c, for more than 20 concurrent G.723.1 flows. It should also be noted that this constant overhead also appears on Fig. 2c for the G.711.1 codec in concurrent state between 5 and 30 flows. This 1.2 s

[5]See http://www.openbach.org

	Pros and cons	Result fidelity		Recommended VoIP use cases
		Satellite QoE	Satellite QoS	
Real satellite access [2]	**Pros:** realism, transparency, advanced satellite features (e.g., performance-enhancing proxies) **Cons:** infrastructure access, shared access for some commercial offers, cost, complex architecture	✓ Mean Opinion Score	✓ Packets dropped ✓ Average jitter ✓ Max. one-way delay	Performance evaluation, tests prior to commercial deployments
Mininet [3]	**Pros:** easy deployment on a single machine, shared user space with virtualized network **Cons:** user should create satellite topology from scratch using hosts, switches and Mininet's API	∽ Mean Opinion Score	✓ Packets dropped ✗ Average jitter ✗ Max. one-way delay	Stress tests
OpenSAND [4]	**Pros:** fine-grained configuration (e.g., carriers, modulation types), emulation of end-to-end satellite communication system with resource allocation **Cons:** several machines required (3), manual management of the whole system	∽ Mean Opinion Score	✓ Packets dropped ✓ Average jitter ∽ Max. one-way delay	Performance evaluation, system dimensioning

TABLE I: Recommendations and comparison of three satellite measurement systems given a satellite VoIP scenario. Legend: ✗: poor approximation; ∽: good approximation under certain conditions; ✓: good approximation.

offset is explained by the difference between each resource allocation schemes. At the time we ran our experiments, we could not access to these confidential vendor characteristics for the real setup. We configured OpenSAND with the most common satellite access method (i.e., static CRA and then dynamic RBDC [13]). In light of the current results, the real satellite access uses a different access method that might rely on a FCA-kind scheme [13]. Despite these differences, OpenSAND clearly exhibits realistic results when focusing on the trends of the results obtained.

For all these reasons, Mininet should be only used for basic VoIP stress tests (when it comes to characterize link capacity for instance) while OpenSAND should be considered for dimensioning. Obviously, for commercial VoIP deployments, real-field satellite tests are inevitable.

V. Conclusions and Perspectives

This paper presented the complexity of emulating a complete end-to-end SATCOM system and the problem linked to performance evaluation. Using a real GEO system, we highlighted the gap of some VoIP results obtained by two commonly-used emulators, namely Mininet and OpenSAND. Following these results, we investigated the root causes of the divergences observed and tried to give recommendations on how to make trustable satellite experiments.

The main objective was to conclude on the consistency of using each setup to drive performance measurements considering the cost to access to a real satellite link.

As future work, we plan to investigate two-way VoIP communications, by using either call traces or SIP phones. Furthermore, we will also attempt to characterize QoE for application services (e.g., video, web browsing) used in a satellite context.

Acknowledgment

The authors would like to thank the CNES for letting us using a real GEO system through the CESARS platform. We also wish to thank Béatrice Paillassa and Gentian Jakllari for their interesting comments. Finally, we also thank David Pradas from Viveris and Patrick Gelard, Emmanuel Dubois and Jérôme Mouëza from CNES for their support.

References

[1] N. Kuhn and E. Lochin, "Network coding and satellites," Working Draft, IETF Secretariat, Internet-Draft, October 2018.

[2] CNES' CESARS Center. [Online]. Available: https://entreprises.cnes.fr/fr/cesars/cesars/

[3] 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*. ACM, 2012, pp. 253–264.

[4] E. e. a. Dubois, "OpenSAND, an open source satcom Emulator," in *Kaconf 2017*, 2017.

[5] ITU-T, "G.107 : The E-model: a computational model for use in transmission planning," *International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T)*, 2015.

[6] L. Carvalho, E. Mota, R. Aguiar, A. F. Lima, and J. N. de Souza, "An E-model implementation for speech quality evaluation in VoIP systems," in *10th IEEE Symposium on Computers and Communications (ISCC'05)*, 2005.

[7] A. Lakaniemi, J. Rosti, and V. I. Raisanen, "Subjective VoIP speech quality evaluation based on network measurements," in *ICC 2001. IEEE International Conference on Communications. Conference Record*, vol. 3, 2001.

[8] G. Sarwar, R. Boreli, and E. Lochin, "On the quality of VoIP with DCCP for satellite communications," *International Journal of Satellite Communications and Networking*, vol. 30, no. 4, pp. 163–180, 2012.

[9] E. Lochin, T. Pérennou, and L. Dairaine, "When should I use network emulation?" *Annals of telecommunications*, vol. 67, no. 5, pp. 247–255, 2012.

[10] G. Baudic, A. Auger, V. Ramiro, and E. Lochin, "HINT: From Network Characterization to Opportunistic Applications," in *Proceedings of the Eleventh ACM Workshop on Challenged Networks*, ser. CHANTS '16. ACM, 2016, pp. 13–18.

[11] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous, "Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT. ACM, 2013, pp. 217–228.

[12] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[13] G. Giambene, *Resource Management in Satellite Networks: Optimization and Cross-Layer Design*, 1st ed. Springer Publishing Company, Incorporated, 2010.