The Triangle Platform for End-to-End Performance Analysis of a 5G Video Transmission Network Slice

Christian Grasso, Raoul Raftopoulos, Giovanni Schembra

DIEEI – University of Catania
V.le Andrea Doria 6, Catania
95125, Italy
christian.grasso@unict.it, raoul@raftopoulos.it, schembra@dieei.unict.it

***Abstract*— The evolution of telecommunications networks towards the fifth generation (5G) of mobile networks is stimulating the interest of both academia and industry in defining frameworks to support design and performance evaluation for each part of them. With this goal, the Fire+ initiative Triangle is developing a pre-5G testbed platform to test, benchmark and certify app, devices and network configurations. The target of this paper is to describe a preliminary experiment to test performance of a video transmission in a radio access network (RAN) using the Triangle Platform. All the steps needed to setup and run the experiment are described in the paper, together with some preliminary numerical results.**

Keywords—5G, RAN, Triangle, Video Transmission, QoS.

I. Introduction

Nowadays, the number of people using streaming video services instead of having a cable subscription is growing significantly. According to YouTube statistics [1], by 2025 half of the viewers under 32 will not subscribe to a pay-TV service. This represents an important challenge for Telco Operators, who must be able to provide users with a network able to manage the increase of the load to deliver services requested by users. 5G network will represent a solution to this kind of Telco Operator's problems.

5G network will be an architectural revolution compared to previous technologies. It will guarantee much better performance, thanks to the increasingly use of cloud technologies and techniques like Software Defined Network (SDN) [2-3] and Network Function Virtualization (NFV) [4-5] to make the network itself more adaptable to the needs of the users. This will allow the Telco Operator to manage not multiple devices and closed systems, but processes and software features that can be moved on distributed hardware.

The main peculiarity of SDN is to decouple the control plane and data plane, using a protocol that modifies forwarding tables in network switches, shifting network intelligence to a centralized SDN controller [6-7]. This controller has a complete view of its status and both switches and applications can communicate with it in real time. In this way the controller allows the network to interact with the applications and, if necessary, to reconfigure itself in an efficient manner.

NFV technology introduces an important change in the approach of network service provision: network functions will become software applications. This allows Telco Operators to migrate functions based on particular optimization policies, to launch more network functions without the problem of having to install new hardware, and give them the ability to modify or fix the behavior of a function simply with a software update, freeing up the operators from intervening directly on the hardware.

Using SDN and NFV paradigms, Telco Operators will be able to reduce both CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX).

The main aspects that will characterize 5G networks are [8]:

- an increase in bandwidth both in uplink and in downlink;
- higher data rate;
- massive connectivity of devices;
- constant supply of QoS;
- lower end-to-end (e2e) latency.

Another key feature of 5G is network slicing [9]. It consists in the creation of a dedicated virtual network architecture with the aim of providing specific functionalities for certain services exploiting the real available physical network. Telco Operators will be able to guarantee streaming services by assigning the user's flow to a particular slice according to the required QoS requirements. So, if a user requires real-time streaming, his flow will be inserted in a slice with very low latency; if he needs high-quality content delivery, his flow will be inserted in another slice.

In this context it becomes important to have tools that allow to test how the network responds to certain requests for services. These tests must involve not only the devices on the network, but also the physical devices of the users and the applications used to request services.

The Triangle Project [10] makes it possible, thanks to a platform developed to test a content distribution service app on a real 5G testbed. The Triangle Project makes available to users, developers or researchers, a 5G network testbed, with the aim of carrying out tests with various network scenarios and focusing their attention on particular aspects of the requested service.

In this paper, we perform a first step in the direction described so far, that is, we will describe how using the Triangle platform to test video transmission focusing on the Radio Access Network (RAN) part of an end-to-end connection. Using the Triangle platform to test network virtualization with SDN and NFV, and multi-access Edge Computing (MEC) [11] facilities is considered as a future work. In Section II it is shown how we setup our experiment,

This work was supported by the EU TRIANGLE project.

the analyzed app, the selected scenario and some obtained numerical results.

II. EXPERIMENT SETUP

The app used to execute the tests was FlyTube (fig. 1), an Android application that allows the user to search for and play YouTube videos in a floating popup window that can be resized and used for multitasking listening experience. An experiment is constituted by the following steps:

1. generate FlyTube app user flow;
2. upload both the application apk and the app user flow on the Triangle Platform;
3. select the features of the app;
4. create and execute the measurement campaign;
5. obtain experiment results.

As we said so far, the first step to run the tests through the testbed is generating the FlyTube app user flow installed on a Samsung Galaxy S4 device. To this purpose, as suggested by the Triangle developers, we used the Quamotion WebDriver tool [12]. Quamotion is a test automation framework for use with native, hybrid and mobile web apps. Quamotion WebDriver is able to automate the actions of users, allowing to manage the entire life cycle of the app (installation, start and automation) using the WebDriver protocol. It automates apps on real devices (iOS, Android), simulators (iOS) and emulators (Android). Only a valid application package (apk or ipa file) and, in the case of iOS, a valid DeveloperProfile and an iOS DeveloperDisks, are needed. Furthermore, a license needs to be requested and uploaded in order to use all Quamotion features. The Quamotion WebDriver design follows the W3C Web Driver specifications. Quamotion WebDriver is an extension of the Web Driver specifications, and adds specific support for managing mobile devices and mobile applications. Scripts can be written in multiple languages (PowerShell, Java, C#). The Developer Options must be enabled on the device to ensure that it could be recognized by the Quamotion WebDriver. The device was then physically connected, via USB cable, to the computer in which Quamotion WebDriver tool is installed.

Clicking on the device icon inside the Device Management section of the Quamotion's Mobile Application Frontend (see Fig. 2.a), lets the user view and download the apk files of all the applications installed on the device (Fig. 2.b). Basic information, such as the serial number of the device, can easily be found. By selecting the device, you can also start a remote screen of it, which allows you to view the screen remotely. The application to be spied is installed in the Quamotion tool by pressing the button next to the apk file name. It is therefore possible to spy the application by clicking on the Inspection App section, which consists of two inspection tools: Spy and Recorder (Fig. 2.c). Using the *Spy*, which is based on the active session and the device's remote display, users are able to click on an element of the screen and obtain information about the widget of the selected user interface. The widget selected is highlighted in the remote screen. Spy then shows the following information:

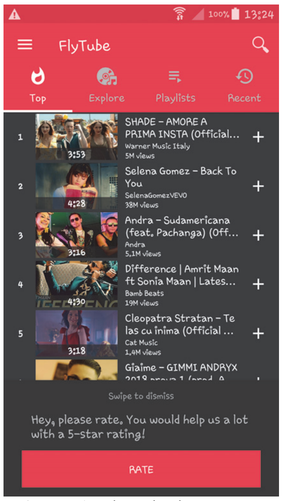
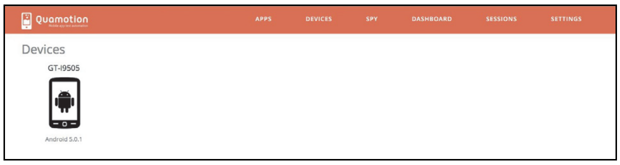
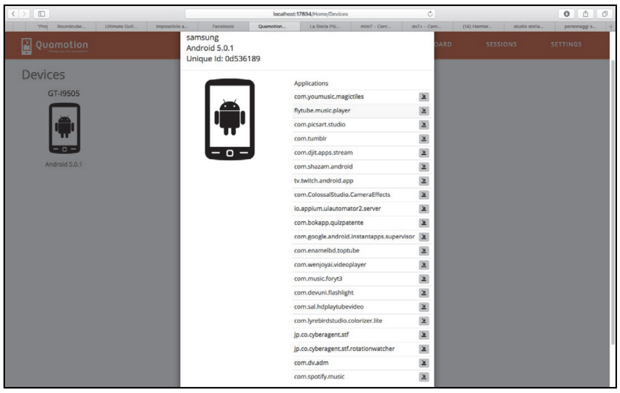


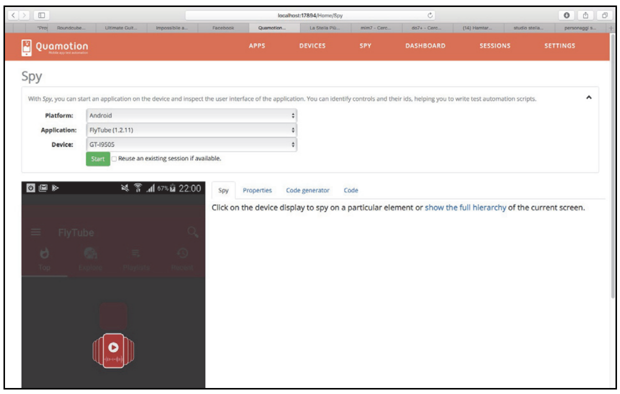
Figure 1. FlyTube home page



(a) Device section of Quamotion WebDriver Tool



(b) View of the installed app on the user smartphone



(c) Spy section of Quamotion WebDriver Tool

Figure 2. Steps for user's flow creation

- Xpath of the widget that uniquely identifies it, which can then be used in the script to search for the elements;
- A tree showing the widget's ancestors;
- All properties of the selected widget.

Recorder generates a PowerShell script based on user interactions on the device, which can also be saved in C#, Java and JSON. Gestures such as touching and inserting text are automatically recorded. The script generated by using Recorder can be subsequently modified wherever necessary.

The script used in all our tests is the following.

1. *Set-TimeOut –time 5000*
2. *Click-Element -xpath "AppCompacttextView [@marked='marked textview'][1]"*
3. *Start-Sleep –Millisecond 3000*
4. *Click-Element -xpath "SuraceView [@marked=surfaceView'][1]"*
5. *Start-Sleep –Millisecond 30000*
6. *Click-Element -xpath "View [@marked=drag_band_view'][1]"*
7. *Start-Sleep –Millisecond 30000*

The script launches the app and selects the first video available inside the Most Viewed Videos section, which will then be displayed in full screen; after 30 seconds, in order to get a buffering, the video is forwarded to the point of half its duration, and after 60 seconds the application is terminated.

In the second step of the experiment setup, the script must be converted and saved in a JSON file in order to be loaded into the Triangle Portal. It is now necessary to upload the apk file of the application to be tested in the testbed portal, in the Apps section. Then, i.e. the third step, the test cases related to the Content Distribution Streaming Services need to be selected; here, we can select one or both available test cases:

- Download content for offline playing
- Non-interactive playback

For each test case, the user flow performing the specific functions of the test cases, such as downloading or playing a video, must be uploaded. The complexity of the testing is wrapped into high-level scenarios, which will prevent users from having to deal with the full set of configurable parameters. A high-level scenario represents network conditions which are configured during test execution.

The fourth step is the creation of the measurement campaign, achieved by selecting the application and its version, the device and the high-level scenario (Fig. 3). Based on this selection, the testbed will configure the physical components and schedule the execution of the tests and the collection of measurements required to check the performance of the features of the application or device under test. The device to be selected is not the one used with the Quamotion tool, but the one that will execute the user flow in the remote testbed in the Triangle platform. All the tests were performed using a Samsung Galaxy S7 and the above mentioned user flow. The three high-level urban scenarios we considered in our experiments are:

- Urban – Office;
- Urban – Internet Cafe - Busy Hours;
- Urban – Driving – Normal.

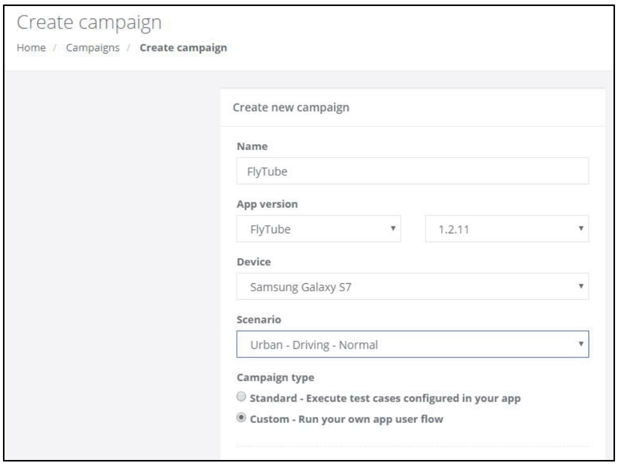


Figure 3: Measurement Campaign Setup

Urban scenarios are characterized by a very dense urban network, in which in addition to the macro sites at a reduced distance from each other, we also find small cells. This dense structure arises from the need to transport a high amount of traffic.

In the *Office* scenario, users will be static, with internal network access points that can be either WiFi or small cells. The characteristics of the channel and the level of interference experienced will also depend on the signals coming from the external macro cells.

The *Internet cafe – Busy Hour* scenario, as for the Office scenario, considers static users. It represents a typical Multiple Radio Access Technology (M-RAT) scenario in which everything occurring at street level (passing by other users and above all moving vehicles) represents a source of interference for the user's channel. The *Busy Hours* option is associated with the Internet Cafe because it is easy to imagine that during the lunch break the place is full of customers who contend for the resources made available by the network.

The *Driving – Normal* scenario fits into the context of a greater number of wireless modems inside the car to communicate with the on-board devices. In the case of Driving Normal scenario it is assumed to have vehicles traveling in urban areas with speeds between 40 and 60 km/h. This makes it reasonable to think a low density of users to represent interference.

If the experiment is successful, the testbed returns to the user three results: measurements, packet capture and log file. About measurements, it consists of a database in which it is possible to visualize different types of values:

- eNodeB information: eNodeB ID, cell ID, physical cell ID, tracking area code, mobile country code, mobile network code;
- system data connection and radio access technology;
- data transmission: RX/TX data and RX/TX data rate;
- quality of signal: RSRP, RSRQ and RSSI;
- smartphone parameters: CPU, RAM, AT4 Battery and average power.

In particular, as an example, we describe results obtained in the Urban Driving Normal scenario with handover, and focus our attention on the following parameters:

- RX Data Rate and TX Data Rate (Fig. 4);
- RSRP and RSSI (Fig. 5);
- eNodeB ID and cell ID (Fig. 6).

Fig. 4a shows the downstream (RX) data rate, while in Figure 4b we can see the upstream (TX) data rate. The video stream, during the experiment, is displayed for 120 seconds but, as seen from the graphic, it arrives at destination in the first fifteen seconds and is buffered: the data transmitted from the core network to the smartphone have been all sent in a single burst.

In the remaining one hundred and five seconds, the RX and TX traffic are only due to signaling traffic. At the instants 56 s and 116 s, the RX and TX traffic record light peaks. In these moments there is the registration of a handover by the user, so in the network there is the relevant exchange of data.

Reference Signal Received Power (RSRP) and Reference Signal Strength Indicator (RSSI) represent the key measurement parameters of signal quality for the modern networks. As defined in [13], RSRP is the linear average over the power contributions (in [W]) of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth. Also in [13], we can find this definition for RSSI: "RSSI is the linear average of the total received power observed only in OFDM symbols carrying reference symbols by UE from all sources, including co-channel non-serving and serving cells, adjacent channel interference and thermal noise". Fig. 5 presents the values of RSRP and RSSI measured during the experiment.

For RSRP, to evaluate signal quality, these ranges are considered:

- [-65, -80] dBm: good signal;
- [-80, -95] dBm: satisfactory signal.

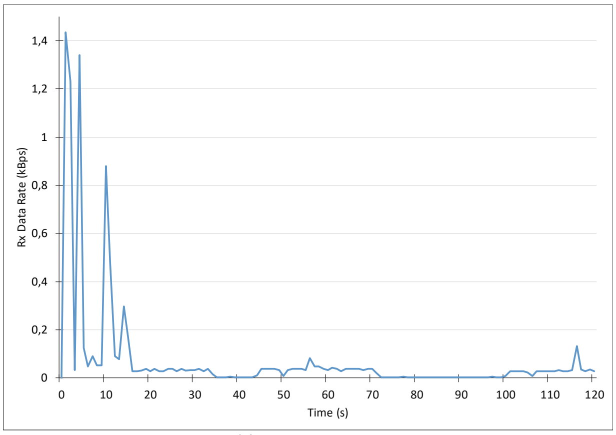
As depicted in Fig. 5, in the considered scenario RSRP values are between -79 dBm and -83 dBm, so signal level is satisfactory.

For RSSI, a good channel is characterized by higher values than RSRP, of at least 25 dB. In the figure, we can observe that RSSI values are 11 dBm higher than RSRP, so this confirms the satisfactory level of the signal.

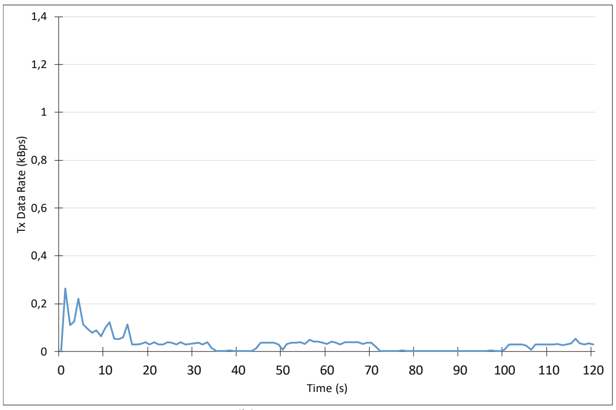
In Fig. 6 we can see measurements related to the Cell ID and eNodeB ID, in order to register movements of the considered mobile user. More specifically, the network performs three consecutive handovers, spaced about of 30 seconds from each other. As we can see in the figure, only the cell ID changes during the experiment, while the eNodeB ID remains the same. This means that the handover performed by the user is an Intra-eNodeB handover: eNodeB's coverage area is divided in sectors each with a different ID (cell ID) and the user moves from a sector to another, remaining in the same eNodeB.

III. CONCLUSIONS AND FUTURE WORK

Measurement and performance evaluation in a real testbed of 5G networks is one of the main challenging actions to support the evolution towards this new network generation. To this purpose, the target of this paper has been the description of a preliminary experiment of video content delivery towards mobile devices in real scenarios. After a



(a) RX Data Rate



(b) TX Data Rate

Figure 4: Measured Data Rate

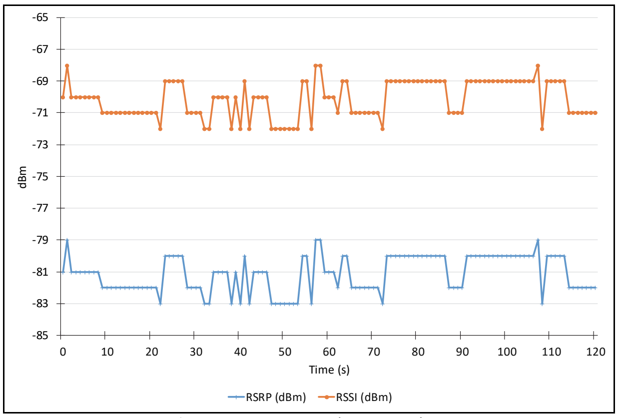


Figure 5: RSRP and RSSI values

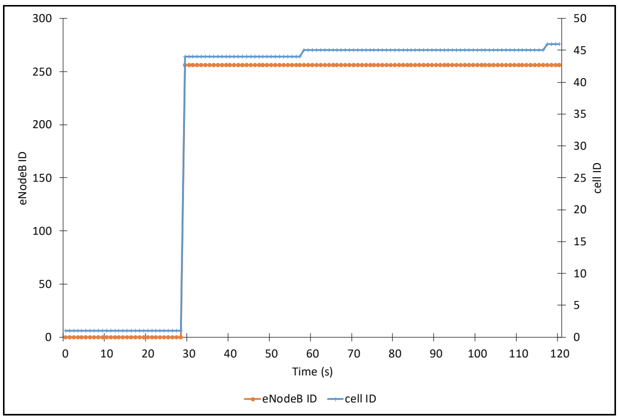


Figure 6: eNodeB ID and cell ID

detailed description of the steps to setup and perform experiments using the Triangle platform, a framework developed within the FIRE+ project Triangle, some preliminary numerical results have been shown as an example of the potentialities provided by that platform in emulating 5G network protocols in real scenarios.

Future work regards the definition of new experiments leveraging the other side of the Triangle platform, that is, the softwarized network provided by an SDN/NFV infrastructure orchestrated by the OSM network orchestrator.

REFERENCES

[1] https://www.youtube.com/yt/about/press/

[2] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky ans S. Uhlig, "Software-defined Networking: A Comprehensive Survey", in *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, January 2015.

[3] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks" in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617-1634, Third Quarter 2014.

[4] S. Abdelwahab, B. Hamdaoui, M. Guizani and T. Znati, "Network function virtualization in 5G", in *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84-91, April 2016.

[5] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta and V. Riccobene, "An open framework to enable NetFATE (Network Functions at the edge)" *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 2015, pp. 1-6.

[6] https://www.opendaylight.org/

[7] https://onosproject.org/

[8] Gupta, Akhil & JHA, Dr RAKESH. (2015). "A Survey of 5G Network: Architecture and Emerging Technologies." Access, IEEE. 3. 1206-1232.

[9] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges." in *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80-87, May 2017.

[10] https://www.triangle-project.eu/

[11] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration", in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017.

[12] http://quamotion.mobi/Mwc/Index

[13] http://www.etsi.org/deliver/etsi_ts/136200_136299/136214/09.01.00_60/ts_136214v090100p.pdf