

An extension of TRIANGLE testbed with model-based testing^{*†}

Laura Panizo, Almudena Díaz and Bruno García

Universidad de Málaga, Andalucía Tech,
Departamento de Lenguajes y Ciencias de la Computación,
Campus de Teatinos s/n, 29071, Málaga, Spain
{laurapanizo,almudiaz,bgarcia}@lcc.uma.es

Traditional testing methods for mobile apps focus on detecting execution errors. However, the evolution of mobile networks towards 5G will require additional support for app developers to ensure also the performance and user-experience. Manual testing in a number of scenarios is not enough to satisfy the expectations of the apps final users. This paper presents the testing framework developed in the TRIANGLE project¹ that integrates a complete mobile network testbed and a model-based testing approach, which is based on model checking, to automatically evaluate the apps performance in different network scenarios.

The TRIANGLE testbed provides an end-to-end mobile network environment that enables app developers to thoroughly test their applications in real network scenarios, including radio conditions, which are not under the control when running tests in live mobile deployments. So the most important feature of the testbed is that the network scenarios emulate realistic conditions, are totally repeatable and the experiments are reproducible.

The integration of model-based testing techniques pursues the automatic generation of app user flows, that represent the interaction of the user with the application, to stimulate the application under test. The integration of model-based testing techniques improves the usability and flexibility of the testbed in different ways:

- The testbed automatically produces a pool of app user flows that use the app features in different ways, improving the test coverage.
- App user flows satisfying different requirements are generated with the same app model. If the testbed is extended with new test cases, the app model does not change, and only new requirements have to be defined.
- The automation of app user flow format is transparent for the app developer. Currently, the portal supports app user flows in JSON format, but we plan to migrate to Powershell scripts in the near future due to new functionality integrated in the testbed.

^{*}This work is funded by the European Union's Horizon 2020 research and innovation programme, grant agreement No 688712 (TRIANGLE project).

[†]This paper has been presented SPIN 2018, and it is published in LNCS vol. 10869

¹<https://www.triangle-project.eu/>