Demo Abstract: Testbed-Independent Experiment Specification and Execution using the COTEFE Platform

Claudio Donzelli  
Telecommunication Networks Group  
Technische Universität Berlin  
donzelli@tkn.tu-berlin.de

Vlado Handziski  
Telecommunication Networks Group  
Technische Universität Berlin  
handziski@tkn.tu-berlin.de

Adam Wolisz  
Telecommunication Networks Group  
Technische Universität Berlin  
awo@ieee.org

Abstract—The demo illustrates the capabilities of the CONET Testbed Federation (COTEFE), focusing on the support for testbed-independent specification and execution of experiments. It leverages a prototypical implementation of the platform and its two core programming interfaces that follow the REST architectural style. The Testbed Abstraction API, that exposes the services of the individual testbeds through a uniform Testing-as-a-Service (TaaS) abstraction, has been implemented and deployed on top of the TWIST testbed. The Testbed Federation API, that leverages these abstractions to build higher-level testbed-independent services, has been implemented and deployed using the Google App Engine infrastructure.

I. INTRODUCTION

The design, implementation and evaluation of cooperating object (CO) protocols and applications is a challenging task that is further complicated by their distributed and heterogeneous nature and the tight coupling with the environment. In the advanced design stages the evaluation of the system performance, error resilience and other nonfunctional properties necessitate use of real hardware, realistic environments and realistic experimental setups.

Testbeds offer convenient middle ground on the realism axes between simulation and full deployment and enable rigorous and controlled experimentation with the System-Under-Test (SUT). Like full deployments, however, they lock the evaluation to one particular environment making it hard to differentiate between the intrinsic properties of the SUT and the influence of the specific features and context at a given testbed site.

One way of decoupling these influences is to cross-validate the functional and non-functional behavior of the SUT under various conditions as provided by different testbeds. Unfortunately, the realization of such measurement campaigns is currently accompanied by significant overheads in configuring the experiments and collecting the results on the individual testbeds, since easy experiment migration is hindered by a lack of common management, experiment specification and control infrastructure.

The CONET Testbed Federation (COTEFE) has been designed to addresses these challenges (Figure 1). It follows a novel Testing-as-a-Service (TaaS) approach that has been adapted to the specific requirements of the cooperating objects domain. Our platform offers convenient access to the resources of multiple testbeds, organized in a federation of autonomous entities. It provides remote services supporting the complete testing life-cycle including resource discovery and reservation, testbed-independent experiment specification, experiment execution and data collection and management.

In the following we briefly outline the main architectural features of the COTEFE platform and the implementation status of its prototype, before presenting the demonstration scenario in greater detail.
II. COTEFE

COTEFE introduces two core APIs which have been designed according to the REST principles [1]. The Testbed Federation API (TFA) answers to the emerging need of supporting design and execution of complex CO experiments and migration across different testbeds by exposing high-level abstraction services enabling experimental research with cooperative objects. The Testbed Abstraction API (TAA) defines the requirements that a CO testbed must fulfill in order to be part of COTEFE. It exposes a selected set of capabilities of the heterogeneous member testbeds under a uniform interface which can coexist with the legacy testbed APIs and allow full autonomy of member testbeds in terms of user management and access policies.

This demo makes use of a prototypical implementation of the two core APIs included in COTEFE which has been developed and deployed for testing. In particular, a prototype of the Testbed Federation API has been implemented and deployed on Google App Engine [2], which enables easy development, deployment and administration of web applications on the Google infrastructure. An instance of the Testbed Abstraction API, based on Django [3], is currently operating on TKN Wireless Indoor Sensor network Testbed (TWIST), the 200+ nodes WSN testbed deployed over four floors in our building at Telecommunication Networks Group (TKN), Technische Universität Berlin. The prototype currently fully supports experiment specification and execution. Visualization of results and debug data is currently supported by a customized Java stand-alone application, which is however based on a testbed-independent configuration script.

Being the two APIs based on RESTful HTTP, every HTTP client (curl, web browsers) or library is a potential client of COTEFE. Two kinds of client have been developed so far: a series of Python scripts (using the modern `httplib2` HTTP library) and a rich HTML5 web interface.

III. DEMO DESCRIPTION

The demo shows the use of the COTEFE APIs for specification and execution of a testbed-independent experiment for cooperating object. The experiment aims to study the behavior of Collection Tree Protocol (CTP) [4] in different interference scenarios. In particular, the experiment focuses on the impact of the interference on the routing topology and the ability of CTP to recover. Interference is activated and disabled by the experimenter at run-time in order to analyze its effects on the network topology. The demo shows how the above mentioned experiment can be performed by interacting with COTEFE in terms of HTTP RESTful APIs. The demo is comprised of three parts:

- **Experiment Specification**: the user describes the experiment using a testbed-independent specification language and uploads its representation to the Testbed Federation server by exclusively using the Testbed Federation API (no testbed is involved here). We show how an experiment can be specified throughout manipulation of REST resources expressed in JSON format. From the specification of required resources, to organization of such resources in subgroups, to the deployment of software images to the different subgroups, and finally the specification of required steps composing the experiment.

- **Job Reservation**: the user uses the created experiment specification (identified by its URL) in order to perform discovery and reservation of required resources on a testbed (in this example TWIST) whose capabilities can satisfy the given requirements. The testbed-independent experiment specification is then translated by COTEFE into a testbed-dependent job specification.

- **Experiment Execution** In this part, the user can perform the experiment by running the corresponding job on the reserved testbed. We show how the experiment can be monitored at runtime and an insight on involved resources can be obtained at any time by submitting a simple HTTP GET requests. In Figure 2 the JSON representation of a WSN node is shown.

IV. CONCLUSION

In this demo we show how a given experiment can be specified in a testbed-independent way and then executed in any of the federated testbeds.

ACKNOWLEDGMENT

This work has been partially supported by CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under the contract number FP7-2007-2-224053.

REFERENCES