



Conference on Networked Systems 2021
(NetSys 2021)

Demo: Simulation-as-a-Service to Benchmark
Opportunistic Networks

Jens Dede, Asanga Udugama and Anna Förster

4 pages

Demo: Simulation-as-a-Service to Benchmark Opportunistic Networks

Jens Dede¹, Asanga Udugama² and Anna Förster³

jd@comnets.uni-bremen.de

adu@comnets.uni-bremen.de

anna.foerster@uni-bremen.de

Sustainable Communication Networks
University of Bremen, Germany

Abstract:

Repeatability, reproducibility, and replicability are essential aspects of experimental and simulation-driven research. Use of benchmarks in such evaluations further assists corroborative performance evaluations. In this work, we present a demonstrator of a simulation service, called "OPS on the bench" which tackles these challenges in performance evaluations of opportunistic networks.

Keywords: Network Simulation, Opportunistic Networks, Benchmarks, OMNeT++

1 Introduction

The confidence in performance evaluations are increased significantly when the underlying simulations are *repeatable*, *reproducible*, and *replicable*, and produce *corroborative* findings [Fei15]. Depending on the complexity of the simulation setup, availability of a comparable baseline and the required computing resources, this is a challenging task.

To address these challenges in evaluations of opportunistic networks (OppNets), we have developed a simulation-as-a-service platform called OPS on the Bench (OOTB) [FKDU21]. It consist of the OppNets model framework [UFDK19] for OMNeT++ combined with a collection of real-world OppNets benchmarks. In this work, we demonstrate the operation of the OOTB platform, simulating OppNets using benchmarks with a variety of data dissemination protocols.

2 Architecture

The work in [KTUF19] shows the variety of evaluations of OppNets. Authors of [KCC05] and [CP16] highlight the problems of such evaluations. Therefore, the OOTB platform, described in this section, is designed to perform *repeatable*, *reproducible*, and *replicable* research to overcome the problems raised in those works.

The OOTB platform uses the **Opportunistic Protocol Simulator (OPS)**¹. OPS [UFDK19] is an **OMNeT++**² based model framework used to simulate OppNets together with the **INET**³

¹ <https://github.com/ComNets-Bremen/OPS>

² <https://omnetpp.org>

³ <https://inet.omnetpp.org>

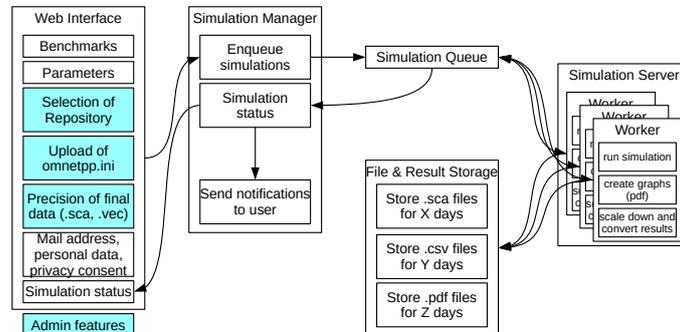


Figure 1: The architectural overview of OOTB

model framework. The models in OPS allow the users to configure OppNet nodes and networks to simulate different scenarios. Each simulation requires a configuration file consisting of the parameter values to run the simulation.

The complete OOTB platform is managed using a **Django Framework**⁴ based application, which offers a user interface, accepts user requests and channels them to backend workers. These workers are implemented using **Docker**⁵ containers and run the simulation itself. The containers are deployed with all required components including the OPS models to run simulations. Once the simulations are completed, a cloud storage service (such as Dropbox) is used to store the results. At the end, the users are informed of the simulation status and the download location of the results.

Figure 1 shows the architecture of OOTB. It consists of five main parts interacting to perform the tasks of accepting simulation requests of users, performing simulations and processing and presenting the results. These parts are described below. A detailed description of the OOTB platform is presented in [FKDU21].

- **Web Interface:** The web interface allow users of OOTB to create and manage simulations. Due to it being browser-based, it is operating system-independent. Access permissions decide the capabilities offered to users, but in general, OOTB offers three possibilities of initiating simulations. Initiating a simulation by (a) invoking a benchmark [FKDU21], (b) using a wizard, or (c) by supplying an OPS / OMNeT++ configuration file.
- **Simulation Manager:** The simulation manager is the central component in OOTB, designed to be the *single point of truth* for all the activities of OOTB. The main tasks are interfacing with the web interface to obtain user requests, enqueueing simulations for the workers in the backend to pick up and simulate and to manage results of simulations.
- **Simulation Queue:** The simulation queue connects the manager and simulation server controlling the multiple workers in the backend.

⁴ <https://www.djangoproject.com>

⁵ <https://www.docker.com>

- **Simulation Server and Workers:** The simulation server runs at least one Docker container – called worker – deployed with the OPS models to simulate OppNets. The worker performs tasks such as monitoring the simulation queue for new jobs, starting new simulations, monitoring active simulations to report status to the manager and to generate summaries of results once the simulation job is complete.
- **Storage (Results & Reports):** The results generated by the server, in the form of scalar results (original OMNeT++ results), summarized results (as CSV files) and plotted results (as .pdf files) are stored in a cloud service for a limited duration for the user to retrieve. General simulation information required to rerun the simulation are stored without a time restriction for later comparison.

3 Running a Simulation

This architecture offers several advantages for users simulating OppNets. The main point is the significantly reduced complexity of running a simulation. Only three steps are important for the user: First Step) Starting a new simulation either by uploading a complete simulation configuration file, setting up the simulation using a wizard (c.f. Figure 2) or changing parameters in a preconfigured benchmark simulation, Second Step) The simulation itself is run automatically on one of the servers without user interaction, Last Step) The user is informed after the simulation has finished. The results including a graphical representation are offered as a download.

The advantages for the user are manifold. Firstly, the setup of the complete simulation environment, including the server, is done by us. Especially for newcomers in this area, this can be a time-consuming task. Secondly, we aim for a standardized representation of the results. This makes the performance of different protocols comparable. It also fosters the reproducibility and comparability in the area of network simulation mentioned at the beginning of this work. Thirdly, with the option of reconfiguring existing benchmark configurations, the impact of the different parameters to a simulation becomes clearer increasing the overall understanding of performance.

4 Demonstration

Start a new simulation

Node Name	Number of nodes	Application Layer Model	Forwarding Layer Model	Wireless Interface	INET mobility
Node Type 1	10	KHeraldApp	KEpidemicRouting	KWirelessInter	Bonn Motion 100
Node Type 2	10	KHeraldApp	KEpidemicRouting	KWirelessInter	Bonn Motion 100

step 2 from 3

first step prev step Continue

Bonn Motion 100 traces
Bonn Motion 200 traces

Figure 2: The second step of the simulation setup: Configure the nodes

We show the main functionality of OOTB in this demonstration: 1) Invocation of the platform, 2) Setting up simulation parameters and commencing simulations, 3) Operations performed in the background by OOTB, 4) Accessing results and comparing with other simulation campaigns. [Figure 2](#) shows a screenshot of the OOTB user interface where the nodes are set up and the corresponding models are selected.

5 Summary and Future Work

OOTB [FKDU21] is a platform to overcome performance evaluation issues in simulations. It is currently available as a preview version⁶ to interested users. Currently, the system is being extended in multiple directions. Optimizing the simulation parsing, better comparison of the results and more flexibility and setup options are some of current work items.

Bibliography

- [CP16] C. Collberg, T. A. Proebsting. Repeatability in Computer Systems Research. *Commun. ACM* 59(3), Feb. 2016.
[doi:10.1145/2812803](https://doi.org/10.1145/2812803)
- [Fei15] D. G. Feitelson. From Repeatability to Reproducibility and Corroboration. *SIGOPS Oper. Syst. Rev.* 49(1):3–11, Jan. 2015.
[doi:10.1145/2723872.2723875](https://doi.org/10.1145/2723872.2723875)
- [FKDU21] A. Förster, T. Karunathilake, J. Dede, A. Udugama. *Benchmarking Data Dissemination Protocols for Opportunistic Networks*. P. 12–19. Association for Computing Machinery, New York, NY, USA, 2021.
[doi:10.1145/3458473.3458819](https://doi.org/10.1145/3458473.3458819)
- [KCC05] S. Kurkowski, T. Camp, M. Colagrosso. MANET Simulation Studies: The Incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.* 9(4), Oct. 2005.
[doi:10.1145/1096166.1096174](https://doi.org/10.1145/1096166.1096174)
- [KTUF19] V. Kuppusamy, U. M. Thantrige, A. Udugama, A. Förster. Evaluating Forwarding Protocols in Opportunistic Networks: Trends, Advances, Challenges and Best Practices. *Future Internet* 11(5), 2019.
[doi:10.3390/fi11050113](https://doi.org/10.3390/fi11050113)
- [UFDK19] A. Udugama, A. Förster, J. Dede, V. Kuppusamy. Simulating Opportunistic Networks with OMNeT++. In Virdis and Kirsche (eds.), *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*. Pp. 425–449. 2019.
[doi:10.1007/978-3-030-12842-5](https://doi.org/10.1007/978-3-030-12842-5)

⁶ <https://ootb.comnets.uni-bremen.de>