



# Enabling iterative development and reproducible evaluation of network protocols



Young-Hwan Kim\*, Alina Quereilhac, Mohamed Amine Larabi, Julien Tribino, Thierry Parmentelat, Thierry Turletti, Walid Dabbous

INRIA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France

## ARTICLE INFO

### Article history:

Received 1 July 2012

Received in revised form 31 December 2013

Accepted 8 January 2014

Available online 23 January 2014

### Keywords:

Reproducible research

Network experiment

Iterative development

Simulation

Emulation

Network testbeds

## ABSTRACT

Over the last two decades several efforts have been made to provide adequate experimental environments, aiming to ease the development of new network protocols and applications. These environments range from network simulators providing a highly controllable evaluation conditions, to live testbeds providing realistic evaluation environment. While these different approaches foster network development in different ways, there is no simple way to gradually transit from one to another, or to combine their strengths to suit particular evaluation needs. We believe that enabling a gradual transition from a pure simulated environment to a pure realistic one, where the researcher can decide which aspects of the environment are realistic and which are controllable, allows improving network solutions by simplifying the problem analysis and resolution.

In this paper, we propose a new network experimentation framework where simulated and real components can be arbitrarily combined to build custom test environments, allowing refining and improving new protocols and applications implementations by gradually increasing the level of realism of the evaluation environment. Moreover, we present a testbed architecture specifically adapted to support the proposed concept, and discuss the design choices we made based on our previous experience in the area of network testbeds. These choices address key issues in network testbed development, such as ease of experimentation, experiment reproducibility, and testbed federation, to enable scaling the size of experiments beyond what a single testbed would allow.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Today, network researchers proposing new protocols and services use many different environments to evaluate their solutions before in the wild deployment. These environments can be generally classified into mathematical modeling, simulation, emulation and live experimentation on real testbeds. Evaluating an operational solution involves tests covering the different parts of the protocol

stack starting from the application, then the transport and network protocols down to the link and physical layers.

It is common to begin the evaluation of a new protocol using mathematical modeling or simulation, and as the protocol design is refined, to switch to more realistic evaluation environment, e.g., first with emulation (e.g. using real application, protocol and OS code and simulated channel model), and later testbed experimentation with real nodes and real traffic.

This staged evaluation process is *costly* in terms of code development (e.g., we might be obliged to re-implement the same protocol twice, once in the simulator and once

\* Corresponding author. Tel.: +33492387825.

E-mail address: [young-hwan@inria.fr](mailto:young-hwan@inria.fr) (Y.-H. Kim).

in the Linux kernel or user space) and usually results in *inaccurate evaluation* (comparison of different algorithms in a simulator and in a testbed might not yield the same results). There is however a need to easily reach a deployable solution whose performance is fairly accurate. Emulab [1] tackles the problem of code reimplementations by using the same scripting language for simulation and emulation, allowing an easy transition from simulation to emulation and vice versa. However, none of existing solutions, including Emulab, provide a fine grained control on the realism level of the evaluation environment.

In this paper, we propose a network experimentation methodology named IDEV for Iterative Development of Network Protocols. IDEV empowers researchers to choose which parts of an experiment should be simulated. Our methodology supports an easy switch from full simulation (all parts including application, protocols, and link layers are simulated) to live experimentation on real nodes where simulations are not used.

In this work, we focus on the evaluation of wireless network protocols to illustrate the complexity of combining simulation and real experimentation. Indeed, the characteristics of wireless links are known to be very variable, unpredictable, and hardly controllable. So, reproducibility of wireless experimentation results is only possible using complex and costly testbeds such as a Faraday Cage shielding Radio Frequency (RF) interference from the outside world and an anechoic chamber to prevent radio waves reflections on the walls.

We propose a set of tools and a testbed architecture to implement the IDEV concept. The proposed framework and testbed leverage on existing technologies, such as the ns-3 network simulator [2], and Direct Code Execution (DCE) [3] for simulation and emulation support, as well as on the OMF wireless framework [4] to support experimentation on real hardware. Our testbed includes 40 high performance wireless nodes and an RF anechoic chamber embedded into a Faraday cage to enable reproducible wireless experimentation results.

The proposed architecture also makes use of the Network Programming Interface (NEPI) [5], to simplify the description, execution and control of the same experiment scenario over different environments. NEPI provides a high-level API to not only interact with resources from different evaluation environments, such as ns-3 and OMF, but also automate network experiment steps including experiment set-up, application installation, and collection of results.

The rest of the paper is organized as follows. In Section 2, we discuss the related work and their differences with our approach. In Section 3, we describe in detail the IDEV methodology and provide several use cases. In Section 4, we present the testbed architecture and infrastructure and we draw conclusions in Section 5.

## 2. Related work

In the network research community, reproducibility is becoming more popular, and some emulation based tools are available to reproduce experimental results. Handigol

et al. [6], replayed thirteen network experiments using Mininet, a container-based network emulator. Mininet enables running standard Unix/Linux network applications and real Linux kernels, in a lightweight and inexpensive way. However, this tool does not allow easily transforming a simulation into an emulation. For example, it is not possible to validate a simulated network module with links emulated in Mininet. Indeed, the gap between simulating a network protocol and emulating it is usually high. Meanwhile, some approaches have been proposed that mix the two worlds: ns-3 [2], Emulab [1], and Flexlab [7].

Ns-3 supports not only simulation, but also emulation using real network links, real applications and network protocol stacks [8,3]. Thus, an experiment using ns-3 can easily be ported between simulation and emulation, by slightly modifying the scenario script.

Emulab uses the ns-2 scripting format to compose network scenarios, thereby it makes it easy to reuse the same script in the Emulab testbed and the ns-2 simulator. However, this tool cannot provide realistic and reliable network conditions. In the Emulab testbed, wired nodes are connected through Ethernet links, but the channel environment is configured and controlled by pre-determined or stochastic channel models, thus the results depend on the reliability of the models. Moreover, wireless nodes are placed in an open space and are exposed to Radio Frequency (RF) noise and interference. Therefore, Emulab is not adequate for reproducible research requiring realistic network conditions.

Flexlab is a hybrid testbed combining some strengths and weaknesses from Emulab and PlanetLab [9]. Although Flexlab is not supported by Emulab since 2007, the approach is remarkable. As mentioned above, Emulab provides full controllability, but the network links are artificially emulated. On the other hand, network links in a real testbed (e.g., PlanetLab) are time-varying as they are connected through the wild Internet. Flexlab provides a portal within the Emulab management system enabling interconnection with PlanetLab nodes. Through the portal, Flexlab can measure network characteristics on real PlanetLab links, and can configure them accordingly into Emulab.

To summarize, none of these approaches allows varying the level of experiment realism in an easy and fine granularity, and this is what IDEV aims to offer.

## 3. Iterative Development (IDEV) of Network Protocols

Network simulation predicts the outcome of real experiments by interactions between the different network entities, such as nodes and protocol stacks. In general, they provide various configurable attributes to evaluate how the network behaves under different conditions. Simulators provide higher flexibility than mathematical models, imitating the behavior of real systems, while still enabling high controllability [10]. However, simulators provide a simplified view of the rules that govern the interaction between network entities, and so the realism that they achieve is limited.

Download English Version:

<https://daneshyari.com/en/article/450820>

Download Persian Version:

<https://daneshyari.com/article/450820>

[Daneshyari.com](https://daneshyari.com)