

# Demonstrating a Unified ICN Development and Evaluation Framework

Alina Quereilhac  
alina.quereilhac@inria.fr

Damien Saucez  
damien.saucez@inria.fr

Priya Mahadevan  
priya.mahadevan@parc.com

Thierry Turletti  
thierry.turletti@inria.fr

Walid Dabbous  
walid.dabbous@inria.fr

## ABSTRACT

Information-Centric Networking solutions target world-wide deployment in the Internet. It is hence necessary to dispose of a development and evaluation environment which enables both controllable and realistic experimentation to thoroughly understand how ICN solutions would behave in real life deployment. In this demonstration, we present an ICN development and evaluation framework that combines emulation and live prototyping environments to provide ICN designers and implementers the means to build “beyond-prototype” ICN solutions. We will demonstrate the benefits of such integrated approach by showing how complete experimental studies can be carried out with minimum manual intervention and experiment set-up overhead, in both emulation and live environments.

## Categories and Subject Descriptors

C.4 [Computer-Communication Networks]: Performance of Systems

## Keywords

ICN evaluation, emulation, NEPI, PlanetLab, ns-3/DCE

## 1. THE NEED FOR EMULATION AND LIVE EVALUATION IN ICN

Simulators and models are at the basis of ICN design and allow to rapidly estimate the impact of new algorithms and strategies. Nevertheless, ICN aims at being deployed in real networks and targets large-scale adoption. In this sense, live deployment offers a rapid feedback on the workability of ICN solutions as it permits to confront hypothesis with the reality. However, since it relies on the usage of real resources, the scalability of live deployment is bounded by physical resource availability. Furthermore, it is also more difficult to debug and isolate faults in real environments due to their uncontrolled nature. Conversely, emulators, offering a less realistic but controllable environment, permit to

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).

ICN'14, Sept 24-26 2014, Paris, France.

ACM 978-1-4503-3206-4/14/09.

<http://dx.doi.org/10.1145/2660129.2660132>

observe the behavior of ICN in larger-scale scenarios or under specific conditions which might be difficult to produce in live environments.

Emulation and live deployment are complementary techniques, as they permit to evaluate a same ICN solution from different angles. However, while it exists a plethora of simulators, emulators, and testbeds to evaluate ICN networking models [2, 9, 3, 5, 10, 4], using any of these approaches in a same study or software development process is non trivial since it requires to manually re-map experiments from one environment to another. We advocate that, for the success of ICN, the community must dispose of a framework to easily use both emulation and real deployment in an integrated and interchangeable way.

Ideally, such framework should reduce to its minimum the burden caused by manual experiment setup (e.g., installing software, configuring the emulator or machines in the testbed), and simplify the transition from emulation to live deployment and vice versa. Furthermore, it should allow easy modeling of a wide range of experiment scenarios, provide a simple way of debugging experiments through adequate instrumentation, and facilitate gathering statistically meaningful data.

## 2. A FRAMEWORK FOR ICN DEVELOPMENT AND EVALUATION

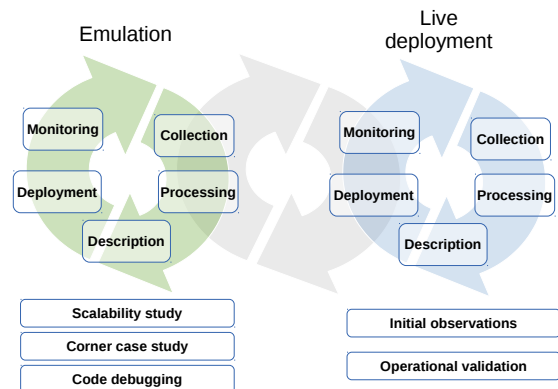
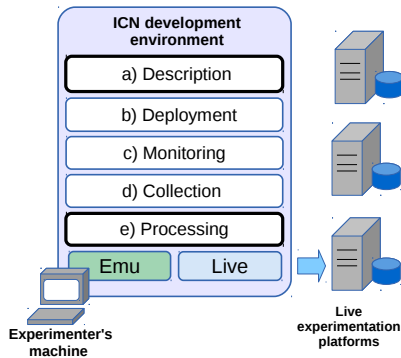


Figure 1: Iterations between Emulation and Live deployment as means of developing ICN software.

The past 30 years of research in the field of the Internet have shown that moving from a lab prototype to a deploy-

able implementation is far from being trivial and requires substantial implementation efforts (coding, testing, and debugging). Emulation and live deployment play complementary roles in the development and evaluation of ICN solutions. Interchanging live deployment and emulation is hence a way to take the best of the two worlds. Both approaches can be combined in a same iterative development process, which can be employed to produce robust ICN software from a lab prototype, as depicted in Fig. 1.

With the objective of simplifying the adoption of such development process, we propose a framework for ICN development featuring adequate abstractions to automate the five fundamental steps of experimentation, independently of the target evaluation environment. These steps are: a) scenario *description*, b) experiment *deployment*, c) experiment *monitoring*, and d) result *collection* and e) *processing*.



**Figure 2: Components of the ICN development and evaluation environment, combining emulation and live evaluation in a unified framework.**

Our framework extends the *Network Experimentation Programming Interface* (NEPI) [7, 1] and leverages on, but is not limited to, the ns-3 simulator tool and its direct code execution emulation extension (DCE) [2, 8], and the live deployment Planetlab testbed [4]. We use DCE as it allows to run real application binaries in ns-3, which means that the exact same code can be used in both the emulation environment as well as in live deployment experiments. Fig. 2 summarizes the architecture of our framework.

From the five experimentation steps, only a) and e) require user intervention, while b), c) and d) are resolved by the framework. Based on a scenario description provided by the user, our framework automates experiment deployment, monitoring and data collection. Statistically meaningful data can be gathered by requesting the framework to automatically repeat steps b) through e) a fixed number of times, or until a criteria specified by the user is met (e.g., convergence of a given metric estimator).

### 3. DEMONSTRATION

In this demonstration we focus on how to implement experiments using our framework. To this end, we first show how to describe experiments independently of the execution environment, using a graph abstraction. We then show how to use NEPI resource modeling primitives to translate an abstract experiment description into an executable scenario for a concrete target environment. We additionally explore the primitives used to deploy and monitor experiments, and

to collect data generated during experiment execution. Furthermore, since one of the usual questions that arises when performing experimental studies is the number of experiment runs to carry out, we show how the framework can be used to automatically re-run an experiment until it determines that the number of runs is sufficient to cover the variability of the studied system. Finally, we show how, thanks to the ability to execute the same application binaries on PlanetLab and DCE environments alike, results generated for a same scenario in either environment have exactly the same format, and thus the obtained data can be processed using the same data processing functions defined by the user. This feature renders comparability of results between emulated and live scenarios feasible.

The demonstration is articulated around three variations of the same content retrieval scenario using CCNx [6]. The first two variants show how, thanks to our framework, it is possible to execute the same scenario in two different environments (i.e., live and emulated), performing only simple changes in the experiment description. The third variant shows how, with almost no extra effort, a scenario can be scaled in size or its topology modified.

All the material presented in this demonstration is available at <http://nepi.inria.fr/>.

### 4. REFERENCES

- [1] Nepi: Network experiment programming interface. <http://nepi.inria.fr>.
- [2] The ns-3 network simulator. <http://www.nsnam.org/>.
- [3] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.
- [4] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. In *SIGCOMM '03*, volume 33, pages 3–12, New York, NY, USA, July 2003. ACM Press.
- [5] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *CoNEXT*, pages 253–264, New York, NY, USA, 2012. ACM.
- [6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. CT, pages 1–12, New York, NY, USA, 2009. ACM.
- [7] A. Quereilhac, C. Freire, M. Lacage, T. Turlletti, and W. Dabbous. Nepi: An integration framework for network experimentation. *SoftCOM*, pages 1–5, 2011.
- [8] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turlletti, and W. Dabbous. Direct code execution: Revisiting library os architecture for reproducible network experiments. *CoNEXT*, pages 217–228, New York, NY, USA, 2013. ACM.
- [9] A. Varga and R. Hornig. An overview of the omnet++ simulation environment. *Simutools*, pages 60:1–60:10, ICST, Brussels, Belgium, 2008.
- [10] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *OSDI*, pages 255–270, Boston, MA, Dec. 2002.