

CCN Simulators: Analysis and Cross-Comparison

Michele Tortelli*, Dario Rossi†, Gennaro Boggia*, Luigi Alfredo Grieco*

*Politecnico di Bari, Bari, Italy - first.last@poliba.it

†Telecom ParisTech, Paris, France - first.last@telecom-paristech.fr

ABSTRACT

This demo focuses on the cross-comparison of CCN simulators available as open source software. The aim is to start a quantitative evaluation of the accuracy, coherence, as well as scalability of software tools available for CCN, in order to understand their boundaries and check if they achieve consistent results. The demo process consists of showing results produced by the tracing systems of each simulator using an interactive parallel coordinate graph, which allows different metrics to be shown at the same time. Both the consistency of simulation results and the differences between several combinations of forwarding strategies, cache replacement policies, and network settings can be verified by users that can interact by proposing and reproducing their own scenario in more than one simulator.

Categories and Subject Descriptors

D.2.4 [Software Engineering]: Software/Program Verification—*validation*; D.2.8 [Software Engineering]: Metrics—*performance measures*.

Keywords

Information Centric Networks; Open-Source Software; Simulation; Performance Evaluation;

1. INTRODUCTION

The relevance that Information Centric Networking (ICN) has gained inside the research community over the last years is confirmed by the number of architectural proposals that go under its umbrella, which are surveyed in [4], as well as by the number of tools (i.e., prototypes, emulators, simulators) released as open source software, which are surveyed in a technical report available at [1].

This demo is motivated by the main findings presented in [1], which complements the qualitative description of the different ICN architectures done in [4]. In particular, in [1] the authors highlight that half of the ICN software tools

are related to a specific architecture, that is Content Centric Networking (CCN) [3], while the other half is divided between 5 different architectures. Furthermore, CCN is the only architecture that presents a complete software ecosystem, including prototypes, emulators and simulators, which represent the majority among them, while most of the ICN tool set is composed by prototypes. This means that, at this stage, the comparison of different strategies within the CCN architecture, combined with a cross-comparison of simulators that are specifically designed for that, is feasible (multiple tools are available) and relevant (increasing attention is dedicated to CCN). A cross-comparison of two or more ICN architectures, instead, would be far more complex and resource expensive, both because of the absence of a software tool that embraces more than a single architecture, and because of the need to set up and run real experiments.

The predominance of simulators as the main software tool for CCN is justified by the fact that, despite the relevance that real experiments have in the standardization process of a new architecture, they guarantee a good compromise between cost and complexity. But even considering only simulators for CCN, only a few common set of features and algorithms is shared between all of them [1]. Therefore, a comparison of different strategies would often require the use of different simulators, whose results need to be validated through a rigorous cross-comparison. Indeed, starting from the implementation of the same scenario in different simulators, the accuracy and the consistency of results can be validated, thus allowing users to choose a specific simulator according to their needs and scenario of interest. Furthermore, the cross-comparison of software tools in terms of accuracy and consistency finds its motivation also in the literature, where, not rarely, discrepancies are found across multiple tools when reproducing the same scenario [2, 5].

The goal of this demo is to provide users a tool to verify the consistency of results produced by different simulators (e.g., by simulating the same scenario in more than one simulator), as well as to compare the performance of different combinations of forwarding strategies, cache replacement policies, cache decision policies, and other parameters. In practice, using an interactive parallel coordinate graphs, which let us show different metrics at the same time, users can: (i) use the mouse to scroll through results obtained using a predefined set of simulations, thus highlighting and showing up statistics of the ones they are interested in; (ii) propose different and new scenarios that will be simulated in real time using a connection with a remote server, on which

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(s). Copyright is held by the author/owner(s).

ICN'14, September 24–26, 2014, Paris, France.

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

<http://dx.doi.org/10.1145/2660129.2660133>.

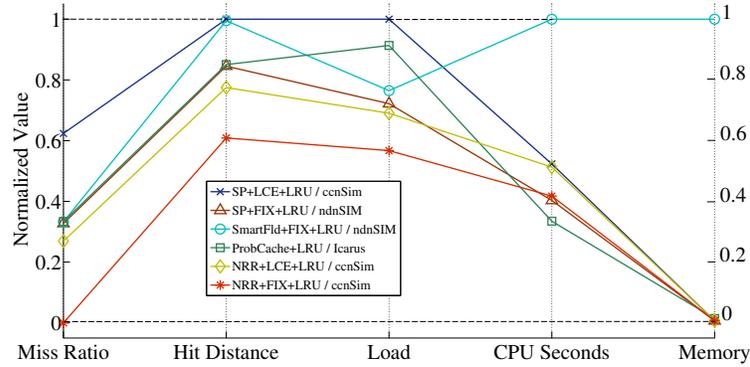


Figure 1: Comparison of the three CCN simulators through different strategies.

all the simulators are installed, thus adding new lines to the interactive parallel coordinate graph.

2. DEMO SOFTWARE

In this section a more detailed description of the demo software is presented, introducing the available simulators, the monitored metrics and the interactive way they are presented to the users.

2.1 Simulators and Parameter Set

After having surveyed the open source software tools for ICN, the authors in [1] provide a brief overview of the CCN simulators, showing their main features in terms of available cache replacement strategies, cache decision policies, forwarding strategies, application levels, and so on. Following their description, three CCN simulators have been chosen to be included inside the demo software. In particular, these simulators are *ndnSIM*, *ccnSim*, *Icarus* (a detailed description is reported in [1]).

During the demo, users can perform simulations by combining different parameters chosen from the main ones reported in Tab. 1; furthermore, they can propose their own scenario by changing other settings, like the simulated network, the content catalog cardinality, the cache to catalog ratio, and so on.

Table 1: Tuneable Features of available CCN Simulators

	Cache Replacement	Decision policy	Forwarding strategy
ndnSIM	LRU,LFU	LCE,FIX	SP
	FIFO,RND		FLD variants BestRoute
ccnSim	LRU,LFU	LCE,FIX	SP
	FIFO,RND	LCD,BTW	FLD variants
		ProbCache	NRR
Icarus	LRU,LFU	LCE,FIX	SP
	FIFO,RND	LCD,BTW	HR variants
		ProbCache,HR	

LCE = Leave Copy Everywhere, FIX=Fixed probability
 LCD=Leave Copy Down, BTW=Betweenness
 HR=HashRouting, FLD=Flooding
 SP=ShortestPath, NRR=Nearest Replica Routing

2.2 Metric Definition and Representation

The performance metrics that the demo software is set to gather from each simulator are representative of the performance of the simulated CCN strategy (i.e., the combination of cache replacement, cache decision policy, and forwarding

strategy), as well as of the performance of the simulators themselves. In particular, they include: mean *miss ratio*, calculated as $1 - \text{hit ratio}$, mean *hit distance*, expressed in number of hops needed to satisfy an Interest packet, and *network load*, expressed in terms of total number of packets (both Interest and Data) generated inside the network. To quantify the performance of each simulator, instead, two metrics are shown: the *CPU seconds* and the *Memory usage*.

The aforementioned metrics will be shown during the demo using an interactive parallel graph, an example of which is reported in Fig. 1. Here the collected metrics are reported after having them normalized with the respective maximum observed values. It is worth to specify that the miss ratio is calculated as $1 - \text{normalized hit ratio}$ (so the most performing strategy could provide a zero miss ratio). Furthermore, a zero memory consumption for some strategies reported in Fig. Fig. 1 is due to the normalization with the far greater memory consumption value of *ndnSIM* in that scenario.

This kind of graph permits to have a view of all the metrics at the same time, and its interactivity relates to the possibility that users have to simulate new scenarios, and/or to highlight particular curves using the mouse, in order to read the corresponding scenario and have information about the main settings related to that scenario.

Acknowledgements

This work has been carried out at LINCS <http://www.lincs.fr>

3. REFERENCES

- [1] <http://telematics.poliba.it/images/file/tortelli/icn-2014.pdf>.
- [2] S. Jansen and A. McGregor. Performance, validation and testing with the network simulation cradle. In *IEEE MASCOTS*, 2006.
- [3] V. Jacobson et al. Networking Named Content. In *ACM CoNEXT*, 2009.
- [4] G. Xylomenos et al. A survey of information-centric networking research. *Communication Surveys and Tutorials, IEEE*, 16(2), 2014.
- [5] A. Stetsko, M. Stehlik, and V. Matyas. Calibrating And Comparing Simulators for Wireless Sensor Networks. In *IEEE MASS*, 2011.