

Dario Rossi

Software::CcnSim

View Edit History Attach Print Logout Stats

RESEARCH

- Curriculum Vitae
- Professional Activities
 - Awards / Committees
- Alumni (and info for prospective students)
- Publications
- Seminar
 - Next talks
 - Archives

TEACHING

- RES203
- INF570
- Projects
- Mobilité Filière RES
- Past

PROJECTS

- Ongoing
 - NewNet@Paris
- Previous

DATASETS

- IP-ID
- WebMOS
- Anycast
- LEDBAT+BitTorrent
- LEDBAT+AQM
- Bufferbloat
- ClassificationDemo
- SkypeDemo
- ICN'14 Demo

SOFTWARE

- ccnSim
- 10Gbps+ tools
- ModelNet-TE
- LEDBAT
- LBE
- wowSim
- BrainDrain

[edit SideBar](#)Acknowledgements
(temporal order)

- ccnSim development started in the context of the [ANR Project Connect](#)
- ...continued thanks to funding from the EIT ICT Labs project on Smart Ubiquitous Content
- ...continued thanks to funding from the EIT ICT Labs project on Virtual Data Plane for Software defined network
- ...and now continues thanks to Cisco's Chair [NewNet@Paris](#)

People (alphabetical order)

- Andrea Araldo (former principal suspect)
- Raffele Chiocchetti (former developer)
- Emilio Leonardi (well informed outsider)
- Dario Rossi (occasional debugger)
- Giuseppe Rossini (former lead developer)
- Michele Tortelli (infty power user)

If you wish to contact us please use the mailing list ccnsim@listes.telecom-paristech.fr

-- and refrain to send me emails in unicast, as my email loss probability is non zero and my email delay reply is anyway heavy-tailed 😊

Overview

ccnSim is a scalable chunk-level simulator of Information and Content Centric Networks (ICN/CCN) that we make available as open-source software to promote cross-comparison in the scientific community. ccnSim is written in C++ under the [Omnet++](#) framework, and features two simulation engines.

- A classic-event driven engine (available in all versions) allows to assess CCN performance in scenarios with large orders of magnitude for CCN content stores (up to 10^6 chunks) and Internet catalog sizes (up to 10^8 files) on off-the-shelf hardware (i.e. a PC with a fair amount of RAM). If you use ccnSim up to v0.3, we ask you to please acknowledge our work by citing [\[ICC-13\]](#) (thanks!)
- ModelGraft, a new hybrid modeling/simulation engine (available starting from v0.4) that allows for unprecedented scalability: with respect to the (highly optimized) execution times of event driven simulation in v0.3, the new technique allow simulation of much larger networks catalogs and content stores on an exiguous amount of RAM and with over 100x reduction of simulation duration. If use ccnSim v0.4 or above, we ask you to please acknowledge our work by citing [\[COMNET-17a\]](#) (thanks!)
- **NEW!** A parallel simulation engine that achieves 100x gain over ModelGraft and thus a 10000x gain over event-driven simulation! The new technique (referred to as CS-POST-MT in [\[PARALLEL-CACHE-17\]](#)) proposes to, insted of slicing nodes of the network over multiple cores, to slice independent portions of the catalog over multiple cores. In contrast to network slicing, which would incur significant MPI overhead, the new technique exhibit an ideal speedup in the number of cores, which justify the above speedups). We have just released the code (see below) and a detailed technical report [\[PARALLEL-CACHE-17\]](#), don't hesitate to tell us what do you think!

You can check the how fast the new version of ccnSim runs when equipped with the [the ModelGraft engine](#) vs the classic event-driven engine on this [YouTube](#) video (that we demonstrated at [\[ICN-16\]](#)). Now, imagine that the same comparison would apply to the parallel simulation engine, which is orders of magnitude faster than ModelGraft!



Download

- **NEW!** We've just (November '17) released the experimental version of ccnSim-v0.4-Parallel
 - We released the code on GitHub <https://github.com/TeamRossi/ccnSim-0.4-Parallel> (sorry no plans to support it as a docker so far)
 - The code implements the parallel CS-POST-MT technique described in the technical report [PARALLEL-CACHE-17]. Have a look and let us know what you think!
- **NEW!** We've just (July '17) released the latest stable version of ccnSim-v0.4 as a docker image hosted on DockerHub!
 - This will spare the hassle of compiling and setting up the environment and allow you to quickly launch your first ccnSim-0.4 simulation so rush to <https://hub.docker.com/r/nonsns/ccnsim-0.4/> !
 - Note: The container does not support the graphical interface. But, trust us, you do not need it anyway 😊
- We've just committed (May '17) the latest stable version of ccnSim-v0.4 on GitHub!
 - This is exactly the same ccnSim-0.4 version that you can download below, just hosted on GitHub
- The latest stable version of ccnSim is ccnSim-v0.4 (May '17) !
 - This new release does not change simulation API, but introduce new significant breakthrough [ITC28b][COMNET-17a][CCN-TR16] in the core simulation environment that allow for very significant memory reduction and CPU speedup.
 - ccnSim-v0.4 is still able to run classic *Event Driven* simulation (as in v0.3, but with significant reduction of the memory footprint due to the use of inversion rejection sampling)
 - ccnSim-v0.4 further allows to run a novel *Monte Carlo* simulation engine (new from v0.4, with significant reduction of the memory footprint and execution time)
 - The *Event Driven* and *Monte Carlo* engines can be seamlessly used, so [v0.3 manual](#) is still valid
 - While we update the 0.4 alpha, see below the FAQs for solution on the compatibility issues you may encounter with the new version 5.0 of omnet++
- As for the the former versions of ccnSim:
 - Really, there is no reason not to use v0.4 which is not only more complete, but also simpler, more modular and faster!
 - Download for v0.3 is still allowed, which features the *Event Driven* simulation engine only. Notice that v0.4 improves the *Event Driven* engine also, so you should consider moving it to this one (which, unless you have modified the code on your own, should be painless as they are fully interoperable)
 - As of september 2015, I have disabled download of older versions (e.g., v0.1 was used in [CAMAD-12], [NOMEN-12], [ICN-12] and v0.2 in [COMCOM-13], [ICN-13], [ICC-13])
 - We extensively benchmarked ccnSim performance and refactored its code, you can find an account of scalability properties of v0.3 in [ICC-13]
 - Example scenarios to reproduce some of our latest publications [ICN-14a] and [ICN-14b] are also available

Version	Source	Download count	Manual	Scenarios
0.4-Parallel (11/2017)	GitGub project page	How do you count downloads on Git?	ccnSim-0.4-Parallel-Manual.pdf	in GitHub , to replicate [PARALLEL-CACHE]
0.4 (05/2017)	ccnSim-0.4.tgz	166	v0.4 manual	Additional scenarios and scripts: ModelGraft sensitivity
0.4alpha2 (02/2016)	ccnSim-0.4alpha2.tgz	377	TBD, please have a look at [CCN-TR16] and v0.3 manual in the meanwhile	Scenarios [CCN-TR16] included in the alpha
0.4alpha (12/2015)	ccnsim-0.4alpha.tgz	238	TBD, please have a look at [CCN-TR16] and v0.3 manual in the meanwhile	Scenarios [CCN-TR16] included in the alpha
0.3 (10/2014)	ccnsim-0.3.tgz	1084	PDF	Cost-Aware [ICN-14a], NRR [ICN-14b]

0.2 (09/2013)	ccnsim- 0.2.tgz	747	PDF	ccnsim-inrr-scripts.tgz (554 downloads)
0.1 (03/2012)	ccnsim- 0.1.zip	1340	PDF	-

FAQ

■ I have troubles installing ccnSim with omnet++ 5.1 (and above)

Unfortunately, this is due to changes in @opp_makemake@ (from the omnet++ changelog: "Support for deep includes (automatically adding each subfolder to the include path) has been dropped, due to being error-prone and having limited usefulness. In projects that used this feature, #include directives need to be updated to include the directory as well."). Fixing this issue is more involved though than just specifying the folders, since there are other modifications introduced by the 5.1 version of omnet++, among which send() and arrived() methods used to generate and process messages. Given that, aside these non-backward compatible changes, no change is relevant for ccnSim, we recommend you to use [omnet++ 5.0](#)

■ I have troubles installing ccnSim with omnet++ 5.0

Upgrade to ccnSim-v4! If you use older version, keeps reading:

We are aware that v5.0 introduced some non-backward-compatible changes. We are updating the code and instructions to be compatible with both omnet++ versions. In the meanwhile, to successfully install ccnSim with omnet++ 5.0, you need to use these slightly modified [ctopology.cc](#) and [ctopology.h](#), and you need to copy [ctopology.h](#) to \$OMNET_DIR/include/omnetpp (instead of the folder commonly used for v4.x)

■ I have troubles installing omnet++ 4.1 with gcc-4.6

Upgrade to ccnSim-v4! If you use older version, keeps reading:

Please refer to <http://www.freeminded.org/index.php/2011/12/omnet-error-during-startup-no-user-interface-cmdenv-tkenv-etc-found/> for a solution (thanks to Cesar A. Bernardini for pointing this out)

■ I have troubles running ccnSim with Tkenv

We are phasing out the support for the graphical interface. But, trust us, you do not need it anyway



The Tkenv should version should works if you properly installed ccnSim-0.4! If you use older version, keeps reading:

If for some reason you need to use the old version, and when running in graphical mode, you encounter an error like the following:

```
Error in module (Client) abilene_network.client[8] (id=23) at event #117, t=5.872724860981:
You forgot to manually add a dup() function to class ccn_interest.
```

The fix is the following: modify the files include/ccn_interest.h and include/ccn_data.h, replacing the line

```
virtual ccn_interest *dup() {return new ccn_interest(*this);}
```

with:

```
virtual ccn_interest *dup() const {return new ccn_interest(*this);}
```

However, we cannot provide support (neither via email, phone, or avian carrier) over the graphical interface. Sorry.

■ I have troubles unpacking the v0.3 archive

We notice that some with some recent linux OS distributions, there may be troubles in unpacking the archive from the command line (though this is not deterministic with distribution version).

- Assuming to have a terminal opened in the same directory where the archive is stored, issue a `tar xzvf ccnsim-0.3.tgz` command.
- In case that fails, `gunzip ccnsim-0.3.tgz; tar xvf ccnsim-0.3.tar` should work for your system.
- We are currently investigating the issue to be fixed in the upcoming 0.4 release -- so if you have any clue/suggestion why this happens and how to fix it, they are welcome !

■ How do I simulate INFORM ?

Short answer, don't. Unfortunately, we missed manpower to sync back *[ICN-13]* in the main ccnSim tree. Fortunately though, ccnSim-v0.3 implements Nearest Replica Routing (NRR)*[ICN-14a]*, that is the best candidate for comparison. So follow suggestion in *[QICN-14]* to setup a sound comparison! and browse the [online interactive demo](#) presented at *[ICN-14e]* to get an idea why this answer should satisfy you.

■ How do I simulate a tree topology?

Since v0.2, tree topologies are included in the default release; additional tree-like topologies (e.g., redundant trees) are available in the companion script set. In case you're still using v0.1, please notice that if you select single shortest path routing and a single repository on any real topology, this will actually induce a chunk-diffusion tree rooted at the repository (though the resulting tree will not be a *binary* tree).

■ How do I play with graph-related properties ?

Originally, we were computing graph related properties directly within ccnSim (the `betweenness centrality()` function in `ctopology.cc`) at the beginning of the simulation. While the method is simplest for the user, it incurs in however a non marginal overhead, as it requires to compute the betweenness centrality of all nodes over and over, so that repeating simulation over the same topology eventually becomes a useless computational overhead. Additionally, betweenness centrality is just one metric, though there are others graph-related properties (e.g., ego-betweenness centrality, or the one we consider in [NOMEN-12], etc.) that could be considered as well, so that this approach was also limited in scope.

We henceforth decided to follow an approach is similar to the one we adopted to adapt the cache size in [NOMEN-12]: i.e., split computation that are related to topological properties that are static over the whole simulation (e.g., betweenness, etc.) from those that need to be taken frequently and possibly evolve over time (eg. routing and forwarding). This is done by specifying a *betweenness* values of all nodes in the `.ini` file, with instructions like:

```
**node[0].betweenness = 1
**node[1].betweenness = 0.4
```

the values of betweenness (of any other similar metrics) can be easily pre-computed with tools such as <http://socnetv.sourceforge.net>. So this is more flexible (as any metric can fit) and less computationally intensive (as the computation is done once for any new scenario).

Publications

- [PARALLEL-CACHE-17] Tortelli, Michele, Rossi, Dario and Leonardi, Emilio, **Parallel Simulation of Very Large-Scale General Cache Networks**. <http://www.enst.fr/~drossi/paper/rossi17parallel-cache.pdf>, November 2017.
- [COMNET-17b] Mansilha, R., Barcellos, M., Leonardi, E. and Rossi, D., **Exploiting Parallelism in Hierarchical Content Stores for High-speed ICN Routers**. *Elsevier Computer Networks*, 125:132-145, September 2017.
- [COMNET-17a] Tortelli, Michele, Rossi, Dario and Leonardi, Emilio, **A Hybrid Methodology for the Performance Evaluation of Internet-scale Cache Networks**. *Elsevier Computer Networks*, 125:146-159, September 2017.
- [TMM-17] Samain, Jacques, Carofiglio, Giovanna, Muscariello, Luca, Papalini, Michele, Sardara, Mauro, Tortelli, Michele and Rossi, Dario, **Dynamic Adaptive Video Streaming: Towards a systematic comparison of ICN and TCP/IP**. *IEEE Transactions on Multimedia*, 19: 2166-2181, October 2017.
- [ITC28b] Tortelli, Michele, Rossi, Dario and Leonardi, Emilio, **ModelGraft: Accurate, Scalable, and Flexible Performance Evaluation of General Cache Networks**. In *ITC28*, September 2016.
- [ITC28a] Araldo, Andrea, Dan, Gyorgy and Rossi, Dario, **Stochastic Dynamic Cache Partitioning for Encrypted Content Delivery**. In *ITC28*, September 2016.
- [NETWORKING-16] Araldo, Andrea, Martignon, Fabio and Rossi, Dario, **Representation Selection Problem: Optimizing Video Delivery through Caching**. In *IFIP Networking*, pages 323-331, May 2016.
- [SIMPAT-16] Tortelli, Michele, Rossi, Dario, Boggia, Gennaro and Grieco, Luigi Alfredo, **ICN software tools: survey and cross-comparison**. *Elsevier Simulation Modelling Practice and Theory*, 63:23-46, April 2016.
- [CCN-TR16] Michele Tortelli, Dario Rossi and Emilio Leonardi, **Model-Graft: Accurate, Scalable and Flexible Analysis of General Cache Networks**. In *Tech. Rep.*, 2016.
- [ICN-15] {R. Mansilha, L. Saino, M. Barcellos, M. Gallo, E. Leonardi, D. Perino and D. Rossi}, **Hierarchical Content Stores in High-speed ICN Routers: Emulation and Prototype Implementation**. In *ACM SIGCOMM Conference on Information-Centric Networking (ICN'15)*, pages 147-156, San Francisco, CA, September 2015.
- [GLOBECOM-14] A. Araldo, M. Mangili, F. Martignon, D. Rossi., **Cost-aware caching: optimizing cache provisioning and object placement in ICN**. In *IEEE Globecom*, pages 1108 - 1113, Austin, Texas, December 2014.
- [ICN-14a] Araldo, Andrea, Rossi, Dario and Martignon, Fabio, **Design and Evaluation of Cost-aware Information Centric Routers**. In *1st ACM SIGCOMM Conference on Information-Centric Networking (ICN-2014)*, pages 147-156, Paris, France, September 2014.
- [ICN-14b] Rossini, Giuseppe and Rossi, Dario, **Coupling caching and forwarding: Benefits, analysis, and implementation**. In *1st ACM SIGCOMM Conference on Information-Centric Networking (ICN-2014)*, pages 127-136, Paris, France, September 2014.
- [ICN-14e] Tortelli, Michele, Rossi, Dario, Boggia, Gennaro and Grieco, Luigi Alfredo, **CCN Simulators: Analysis and Cross-Comparison**. In *1st ACM SIGCOMM Conference on Information-Centric Networking (ICN-2014), Demo Session*, pages 197-198, Paris, France, September 2014.
- [ICN-14c] Imbrenda, Claudio, Muscariello, Luca and Rossi, Dario, **Analyzing Cacheable Traffic in ISP Access Networks for Micro CDN applications via Content-Centric Networking**. In *1st ACM SIGCOMM Conference on Information-Centric Networking (ICN-2014)*, pages 57-66, Paris, France, September 2014.
- [ICN-14d] Imbrenda, Claudio, Muscariello, Luca and Rossi, Dario, **Analyzing Cacheability in the Access Network with HACKSAW**. In *1st ACM SIGCOMM Conference on Information-Centric Networking (ICN-2014), Demo Session*, pages 201-202, Paris, France, September 2014.
- [INFOCOM-14] G. Rossini, D. Rossi, G. Garetto and E. Leonardi, **Multi-Terabyte and Multi-Gbps Information Centric Routers**. In *IEEE INFOCOM*, Toronto, Canada, Apr 2014.
- [PATENT-08] D. Rossi and G. Rossini, **Method for managing packets in a network of Information Centric Networking (ICN) nodes**. Patent EPO14305866.7, 2014.

19. [PATENT-07] Giuseppe Rossini, Dario Rossi, Michele Garetto and Emilio Leonardi , **Information Centric Networking (ICN) router** . Patent EPO14305639.8, 2014.
20. [QICN-14] Tortelli, Michele, Rossi, Dario, Boggia, Gennaro and Grieco, Luigi Alfredo, **Pedestrian Crossing: The Long and Winding Road toward Fair Cross-comparison of ICN Quality** . In *International Workshop on Quality, Reliability, and Security in Information-Centric Networking (Q-ICN)*, Rhodes, Greece, August 2014.
21. [ICC-13] Chiocchetti, Raffaele, Rossi, Dario and Rossini, Giuseppe, **ccnSim: an Highly Scalable CCN Simulator** . In *IEEE International Conference on Communications (ICC)*, june 2013.
22. [COMCOM-13] Giuseppe Rossini and Dario Rossi, **Evaluating CCN multi-path interest forwarding strategies** . *Elsevier Computer Communication, SI on Information Centric Networking*, 36 : 771-778 , Avril 2013.
23. [ICN-13] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, G. Rossini, **INFORM: a Dynamic Interest Forwarding Mechanism for Information Centric Networking** . In *ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, Hong Kong, China , 2013.
24. [PATENT-06] D. Perino, G. Carofiglio, D. Rossi and G. Rossini, **Dynamic Interest Forwarding Mechanism for Information Centric Networking** . Patent EPO13306124.2 , 2013.
25. [PATENT-05] D. Perino, G. Carofiglio, R. Chiocchetti, D. Rossi and G. Rossini, **Device and method for organizing forwarding information in nodes of a content centric networking** . Patent EPO13161714.4, 2013.
26. [CAMAD-12] G. Rossini, D. Rossi, **A dive into the caching performance of Content Centric Networking** . In *IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'12)*, pages 105-109, september 2012.
27. [ALGOTEL-12] G. Rossini and D. Rossi, **Large scale simulation of CCN networks** . In *Algotel 2012* , La Grande Motte, France, May 2012.
28. [IETF83-a] Dario Rossi, **A dive into CCN's caching performance** . In *Talk at IETF83, ICNRG meeting*, Mar. 31 2012.
29. [NOMEN-12] D. Rossi and G. Rossini, **On sizing CCN content stores by exploiting topological information** . In *IEEE INFOCOM, NOMEN Workshop* , pages 280-285, Orlando, FL, March 25-30 2012.
30. [ICN-12] Raffaele Chiocchetti, Dario Rossi, Giuseppe Rossini, Giovanna Carofiglio and Diego Perino,, **Exploit the known or explore the unknown: Hamlet-like doubts in ICN** . In *ACM SIGCOMM, ICN Workshop*, 2012.
31. [CCN-TR11a] D. Rossi, G. Rossini, **Caching performance of content centric networks under multi-path routing (and more)** . Technical report, *Telecom ParisTech*, 2011.
32. [COMLET-13] Giuseppe Rossini and Dario Rossi, **Exploiting Topology Knowledge in Information Centric Networks: Guidelines and Challenges** . *IEEE Multimedia Communications Letter*.