

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What is ccnSim? . . . . .	2
1.2	Organization of this manual . . . . .	2
<b>2</b>	<b>Downloading and installing ccnSim</b>	<b>2</b>
<b>3</b>	<b>Overview</b>	<b>3</b>
3.1	Overall structure of ccnSim . . . . .	3
3.2	Nodes, caches, and strategy layers . . . . .	3
3.2.1	Core Layer . . . . .	4
3.2.2	Strategy Layer . . . . .	4
3.2.3	Content Store . . . . .	4
3.3	The clients . . . . .	4
3.4	The content distribution . . . . .	4
3.5	Statistics . . . . .	4
3.6	Global parameters . . . . .	4
<b>4</b>	<b>Run your first simulation</b>	<b>4</b>
<b>5</b>	<b>Extending ccnSim</b>	<b>4</b>

# 1 Introduction

## 1.1 What is ccnSim?

ccnSim is a scalable chunk-level simulator of Content Centric Networks (CCN)[1], that we developed in the context of the ANR Project Connect.

- It is written in C++ under the Omnet++ framework.
- It allows to simulate CCN networks in scenarios with large orders of magnitude.
- It is distributed as free software, downloadable at <http://site>.

ccnSim extends Omnet++ as to provide a modular environment in order to simulate CCN networks. Mainly, ccnSim models the forwarding aspects of a CCN network, namely the caching strategies, and the forwarding layer of a CCN node. However, it is fairly modular, and simple to . We hope that you enjoy ccnSim in which case we ask you to please cite our paper [?]. ccnSim is able to simulate content stores up to 106 chunks and catalog sizes up to 108 files in a reasonable time.

## 1.2 Organization of this manual

This manual is organized as follows:

- Organization.

# 2 Downloading and installing ccnSim

You can freely download ccnSim from the project site: <http://code.google.com/p/ccnsim/>.

Moreover, we assume that you have downloaded and installed Omnetpp (version  $\geq 4.1$ ) on your machine. In order to install ccnSim, it is first necessary to patch Omnetpp. Then you can compile the ccnSim sources. These steps are as follows:

```
john:~$ cd CCNSIM_DIR
john:CCNSIM_DIR$ cp ./patch/ctopology.h OMNET_DIR/include/
john:CCNSIM_DIR$ cp ./patch/ctopology.cc OMNET_DIR/src/sim
john:CCNSIM_DIR$ cd OMNET_DIR && make && cd CCNSIM
john:CCNSIM_DIR$ ./scripts/makemake
john:CCNSIM_DIR$ make
```

We suppose that CCNSIM\_DIR and OMNET\_DIR contain the installation directory of ccnSim and Omnet++ respectively.

## 3 Overview

### 3.1 Overall structure of ccnSim

In order to better understand the organization of `ccnSim`, the best is to look at its internal organization. In the following we reproduce the basic directory organization of `ccnSim`.

```
|-- topologies
|-- modules
|   |-- clients
|   |-- content
|   |-- node
|       |-- cache
|       |-- strategy
|   |-- statistics
|-- packets
|-- include
|-- src
|   |-- clients
|   |-- content
|   |-- node
|       |-- cache
|       |-- strategy
|   |-- statistics
```

As said within the introduction, `ccnSim` is a package built over the top of Omnet++. As such, we can divide its implementation in two different subunits. One subunit is represented by the `.ned` description of the modules used by `ccnSim`, and included within the directory `modules` and `topologies`. The first directory, is basically the `.ned` description of the modules employed by `ccnSim`, like clients, nodes, and so forth. Within the `topologies` directory there are some sample topologies (always described in `.ned` format) ready to be used.

The real implementation of these modules lie within the `src` and `include` directory, that contain sources and header files, respectively. Within the rest of this section we summarily describe the features of these components, together with a brief overview of their most important parameters.

### 3.2 Nodes, caches, and strategy layers

Nodes are the core part of the `ccnSim`. They form the core network and can be connected each other by the means of `faces`(see [1] for better understanding what a *face* is). However, nodes own not a real C++ implementation. A node is just a compound module, composed by three parts: the Core Layer, the Strategy Layer, and the Content Store.

### 3.2.1 Core Layer

The core layer implements the basic tasks of a CCN node. Indeed, it handles the PIT, sending data back to the interested interfaces. It handles the incoming interest by replying to it (in the case of a *cache hit*), or by appending the interest to the existent PIT entry. In the case no entry exists yet it dispatches it to the *Strategy layer* in order to get a decision about where (i.e., on which interface) sending it.

### 3.2.2 Strategy Layer

The strategy layer receives an interest for which no PIT entry exists yet. Then, it has to *decide* where sending the given interest over the network.

### 3.2.3 Content Store

## 3.3 The clients

## 3.4 The content distribution

## 3.5 Statistics

## 3.6 Global parameters

# 4 Run your first simulation

# 5 Extending ccnSim

## References

- [1] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking named content. In *ACM CoNEXT*, page 112, 2009.