## NDN Codebase Overview

### Infrastructure Software
- NFD
- NFD-android
- NDN-RIOT
- μNFD
- NDN Tools
- NLSR
- Repo-ng, repo-sql
- NDN Control Center

### NDN Libraries
- ndn-cxx
- NDN-CCL (JS, Java, Python)
- NAC
- ChronoSync
- PSync
- Vector Sync
- NDN-RTC

### Apps
- ChronoChat
- ndns
- ndncert
- ndn-flow
- NdnCon
- ndn-fs
- ndn-atmos
- Many others

### Evaluation Frameworks
- ndnSIM
- miniNDN
- NDN Testbed
Starting Point
Where to Find Source Code for NDN Codebase

• Github organizations
  • [https://github.com/named-data](https://github.com/named-data)
    • NFD, core libraries, and other general use software
  • [https://github.com/named-data-mobile](https://github.com/named-data-mobile)
    • Android and related software
  • [https://github.com/named-data-iot](https://github.com/named-data-iot)
    • IoT related software
  • [https://github.com/named-data-ndnsim](https://github.com/named-data-ndnsim)
    • ndnSIM core, example and real simulation scenarios
Supported Platforms

• Desktop Systems
  • Ubuntu, OSX, FreeBSD and other Linux distributions
• Home routers
  • OpenWRT, DD-WRT
• Mobile:
  • Android, iOS (library only)
• IoT:
  • Arduino, ESP8266, RIOT-OS
  • Raspberry Pi (runs NFD, available binary)
• Web browser
  • NDN-JS library + microforwarder

https://redmine.named-data.net/projects/nfd/wiki
NDN Forwarding Daemon (NFD)

• The reference implementation of NDN forwarder
• [https://named-data.net/doc/NFD/current/](https://named-data.net/doc/NFD/current/)
  • Overview
  • Getting started
  • NFD Developer’s Guide
  • Manpages
  • Wiki
  • API documentation (doxygen)

• Feedback, suggestions, and contributions are welcome.
NDN-Android: NDN Stack for Android

- Embeds actual NFD, compiled using NDK
- Works with all (non-rooted) Android devices

https://play.google.com/store/apps/details?id=net.named_data.nfd
&hl=en
https://github.com/named-data-mobile
NDN-RIOT: NDN for RIOT-OS

- Optimized for IoT apps
- Support
  - Data-centric security
  - Stateful NDN packet forwarding
  - Replaceable forwarding strategies
  - 802.15.4 and Ethernet
- Simple application APIs
- Several simple examples to get started

https://github.com/named-data-iot
NDN Tools

- ndnping, ndnpingserver
  - Rechability testing tools
- ndncatchunks, ndnputchunks
  - Segemented file transfer between a consumer and producer
- ndnpeek, ndnpoke
  - Transmit a single packet between a consumer and a producer
- ndndump, dissect, wireshark-dissect
  - Debug NDN packet flow

- repo-ng, repo-sql: NDN repositories providing managed persistent storage
App Codebases

NFD: NDN Stack
https://github.com/named-data/NFD

NDN Sync
https://github.com/named-data/ChronoSync
https://github.com/named-data/PSync

NDN Repo
https://github.com/named-data/repo-ng

NAC: Name-Based Access Control
https://github.com/named-data/name-based-access-control

ndn-cxx library and security tools
https://github.com/named-data/ndn-cxx

NDNCERT: automated certificate provisioning
https://github.com/named-data/ndncert

NDN tools: Maintenance and Debugging
https://github.com/named-data/ndn-tools
ndn-cxx: NDN C++ library with eXperimental eXtensions

- C++14
- The reference library and security library implementation
- Used in: NFD, NLSR, ndn-tools, ChronoChat, etc.
- [https://named-data.net/doc/ndn-cxx/current/](https://named-data.net/doc/ndn-cxx/current/)
  - Overview
  - Getting started
  - Trivial applications
  - Tutorials
  - Specifications
  - Manpages
  - API documentation (doxygen)
- Feedback, suggestions, and contributions are welcome.
NDN Common Client Libraries (NDN-CPP, NDN-JS, jNDN, PyNDN)

- C++, Java, Python, JavaScript, C#, Squirrel
- Used in: NDN-RTC, NdnCon, NFD-Android, etc.
- [https://named-data.net/codebase/platform/ndn-ccl/](https://named-data.net/codebase/platform/ndn-ccl/)
  - NDN Common Client Libraries API
  - NDN-CPP API
  - PyNDN API
  - NDN-JS API
  - jNDN API
NDN Testbed

• Network of 46 sites across 4 continents, 14 countries
• Open to join and use
• [https://named-data.net/ndn-testbed/policies-connecting-nodes-ndn-testbed/](https://named-data.net/ndn-testbed/policies-connecting-nodes-ndn-testbed/)

• Examples applications and experiments: videoconferencing, network management, virtual machine migration, strategies, nTorrent, etc.

• Small scale evaluations
Open Network Lab (ONL)

- Remotely accessible network testbed
  - Operated and maintained by Applied Research Lab in Department of Computer Science and Engineering at Washington University in St. Louis
  - Real Hardware for running repeatable network experiments with trusted results. (NOT simulations)

- Use for NDN
  - NDN installed on each host/VM
  - NFD performance study
  - NDN Testbed Emulation to test new releases

- To join
  - [https://onl.wustl.edu/](https://onl.wustl.edu/)
    - And “Get an account”
MiniNDN: NDN Emulation Framework (Based on MiniNet)

**Features**
- GUI and experimental framework provided to configure topologies and experiments
- Mini-NDN v0.3.0 released with cluster edition to improve scalability

**Need for Mini-NDN**
- Shared emulation environment (ex: Emulab)
  - Problems with many users, long wait time
- Simulation environments (ex: ndnSIM)
  - Real world applications may need to be modified
  - NFD version may not be the very latest commit on Github
  - Cannot interact directly in real time (visualization is possible)

**Introducing Mini-NDN**, Mininet based network emulator
- Runs actual instances of NFD, NLSR
- Medium-scale evaluations
  - Easy to configure network emulation
  - Runs any real application
  - Number of emulated nodes $\propto$ CPU power
  - Cluster edition can be used to scale emulations

http://minindn.memphis.edu/
ndnSIM: NDN Simulation Framework (Based on NS-3)

- Fully integrated with NDN prototype implementations: NFD & ndn-cxx
- Large scale evaluations
- Provide interoperability between simulation and prototyping
- Enable a two-way of experimentation and evaluation
- Enable high-fidelity NDN simulations
- 1500+ nodes with WiFi channels in the evaluation of NDN for vehicular networking

https://ndnsim.net/