Named Data Networking Daemon

Varun Patil

NDN in Golang

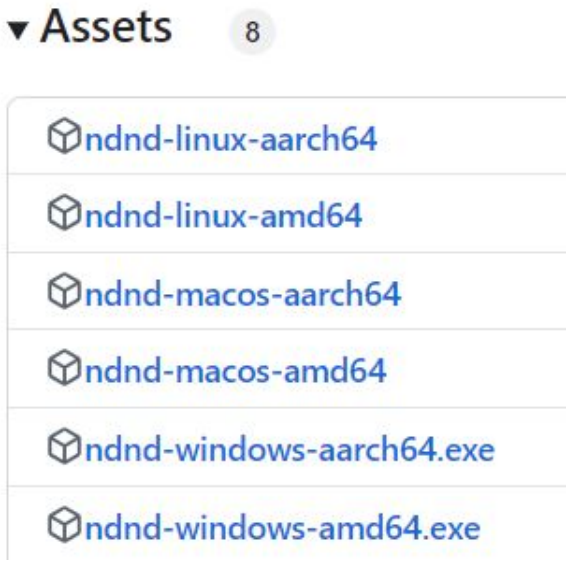
- https://github.com/named-data/ndnd
- Complete NDN implementation in Go
  - Standard Library
  - Forwarder (YaNFD)
  - Routing Daemon (ndn-dv)
  - Tooling
- Permissive License (MIT)

Design Goals

- Consolidate
  - One repo with all basic ingredients
  - Make it easier to find what you need
- Focus on usability
  - For developers: High level APIs and examples
    - SVS-PS
    - LightVerSec
  - For operators: Reduce barrier to entry
- De-generalized / opinionated

Single binary, broad platform support

- Go supports most modern targets
  - Tested on Linux / Mac / Windows
  - Static binaries (zero dependencies)
- WebAssembly
  - Support compilation with TinyGo (standard library only)
- https://github.com/named-data/ndnd/releases
  - Pre-built binaries for many platforms
- https://github.com/named-data/ndnd/pkgs/container/ndnd
  - Pre-built docker images for Linux



```
root@0037b98ec2ac:~# ndnd
  _   _ ____  _   _     _
 | \ | |  _ \| \ | | __| |
 |  \| | | | |  \| |/ _  |
 | |\  | |_| | |\  | (_| |
 |_| \_|____/|_| \_|\____|

Named Data Networking Daemon

Usage:
  ndnd [command]

NDN Daemons
  fw           NDN Forwarding Daemon
  dv           NDN Distance Vector Daemon
  daemon       NDN Combined Daemon

Security Tools
  sec          NDN Security Utilities
  certcli      NDNCERT Certificate Client

Debug Tools
  ping         Send Interests to a ping server
  pingserver   Start a ping server under a name prefix
  cat          Retrieve object under a name prefix
  put          Publish data under a name prefix

Additional Commands:
  help         Help about any command

Flags:
  -v, --version   version for ndnd
```

NDNd packages

- ndnd/fw
  - NDN Forwarder (originally YaNFD)
  - Compatible with NFD, supports most common functionality
- ndnd/dv
  - Routing Daemon (ndn-dv)
  - Compatible with ndnd/fw and NFD
- ndnd/std
  - Standard library to build NDN apps
  - Both low and high level APIs
- ndnd/tools
  - Debugging tools

Combined NDN Daemon

- Single binary / command to run routing and forwarding daemons
- Make it easier to get started with running an NDN network
  - Tutorial available
  - https://github.com/named-data/ndnd/blob/main/docs/daemon-example.md

High Level APIs - Network

- Produce and consume Named Application Data Units (ADUs)
  - Multiple local storage options
- Internally handles data segmentation
- Internally handles signing and validation
- Provide security policy and trust anchor
  - Simplified keychain implementation
  - LightVerSec for trust schema definition

High Level APIs - Example Producer

```
app := engine.NewBasicEngine(engine.NewDefaultFace())
app.Start()

cli := object.NewClient(app, object.NewMemoryStore(), nil)
cli.Start()

name, _ := enc.NameFromStr("/my/ndn/data")
content :=  make([]byte, 819200)

cli.Produce(ndn.ProduceArgs{
    Name:    name.WithVersion(5),
    Content: content,
})
```

High Level APIs - Example Consumer

```
1   app := engine.NewBasicEngine(engine.NewDefaultFace())
2   app.Start()
3
4   cli := object.NewClient(app, object.NewMemoryStore(), nil)
5   cli.Start()
6
7   fetchName, _ := enc.NameFromStr("/my/ndn/data/v=5")
8
9   cli.Consume(fetchName, func(status ndn.ConsumeState) {
10      content := status.Content()
11  })
```

High Level APIs - SVS-PS

- Familiar Publish-Subscribe API
  - **`SvsPS(<group>, <name>, <security-config>)`**
  - **`publish(<name>, <content>)`**
  - **`subscribe(<prefix>, <callback>)`**
- Subscribe to producer prefixes
- Subscribe to data name prefixes
  - (implementation work in progress)
- Let application decide what to fetch
  - Callback with data name

High Level APIs - SVS-PS

- Hide network and transport layer details from app
  - New data notification with Sync
  - Object fetching, congestion control
  - Retransmission if lossy
- Hide signing, verification and policy execution
  - App only provides security policy
  - NDN library executes it

A Simpler Keychain

```
root@0037b98ec2ac:/work/ndnd# ndnd sec keygen
Error: requires at least 2 arg(s), only received 0
Usage:
  ndnd sec keygen IDENTITY KEY-TYPE [params]

Examples:
  ndnd sec keygen /alice ed25519
  ndnd sec keygen /ndn/bob rsa 2048
  ndnd sec keygen /carol ecc secp256r1

root@0037b98ec2ac:/work/ndnd# ndnd sec keygen /alice ed25519

-----BEGIN NDN KEY-----
Name: /alice/KEY/%AEG%8C%B0%86%E1%17%E7
SigType: Ed25519

BrAHFggFYWxpY2UIA0tFWQgIrkeMsIbhF+cUAxgBCRUwMC4CAQAwBQYDK2VwBCIE
IOI3ox9BKtEDoTvjPEcuRaDbnmLDGizD5i83HW2DRzPaFh0bAQUcGAcWCAVhbGlj
ZQgDS0VZCAiuR4ywhuEX5xdAvqFFtzUJ02TiDvbWUtTwq9ixqsGnNQ3SyyE1hPKk
ScB+FmsGbW2ZOH6hEIN/7bsNkfkH3G5UKbNLZxyKKJF0Cg==
-----END NDN KEY-----
```

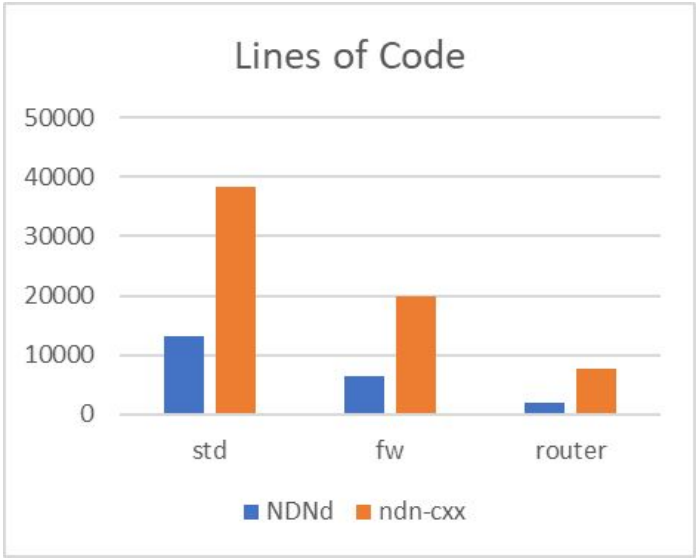
A Simpler Keychain

```
root@0037b98ec2ac:/work/ndnd# ndnd sec sign-cert alice.key --issuer SELF < alice.key

-----BEGIN NDN CERT-----
Name: /alice/KEY/wv%12%5D5%8C%14%C5/SELF/v=1744485119124
SigType: Ed25519
SignerKey: /alice/KEY/wv%12%5D5%8C%14%C5
Validity: 2025-04-12 19:11:59 +0000 UTC - 2026-04-12 19:11:59 +0000 UTC

BuwHJggFYWxpY2UIA0tFWQgId3YSXTWMFMUIBFNFTEY2CAAAAZYranSUFAkYAQIZ
BAA27oAVLDAqMAUGAytlcAMhAEfK321FjwQLbIVzrL7GsWGDauUrc2RVBnEZKmXE
PjIxFkcbAQUcGAcWCAVhbGljZQgDS0VZCAh3dhJdNYwUxf0A/Sb9AP4PMjAyNTA0
MTJUMTkxMTU5/QD/DzIwMjYwNDEyVDE5MTE1ORdALZiVnlD27cUJpR839GEsFOWZ
KZVmJ4P9gLWRUKjFHBtjfCkm8m8j0C9KHddX7m3ofkAvwulQn6EEB6EdS/qdAA==
-----END NDN CERT-----
```

Size of Codebase

- Excluding lines from
  - Examples
  - Tests
  - Generated code
  - Comments / blank lines

	NDNd	ndn-cxx family
Client Library	13165	38350
Forwarder	6392	19824
Router	1880	7623



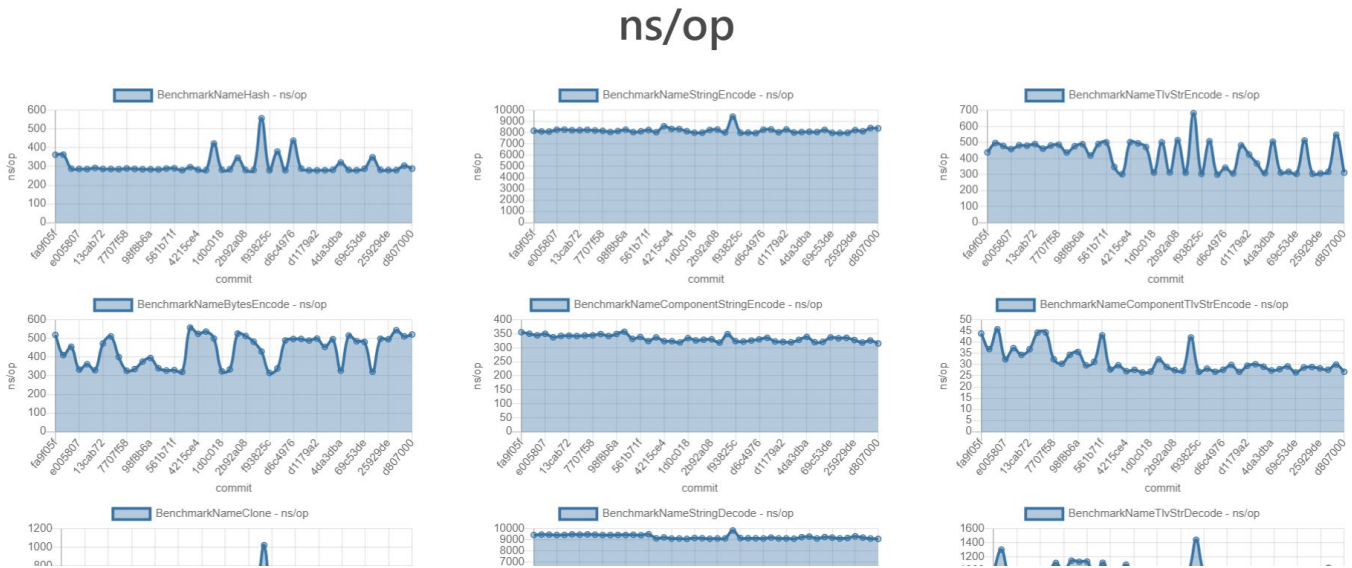
E2E testing with MiniNDN

https://github.com/named-data/ndnd/tree/main/e2e

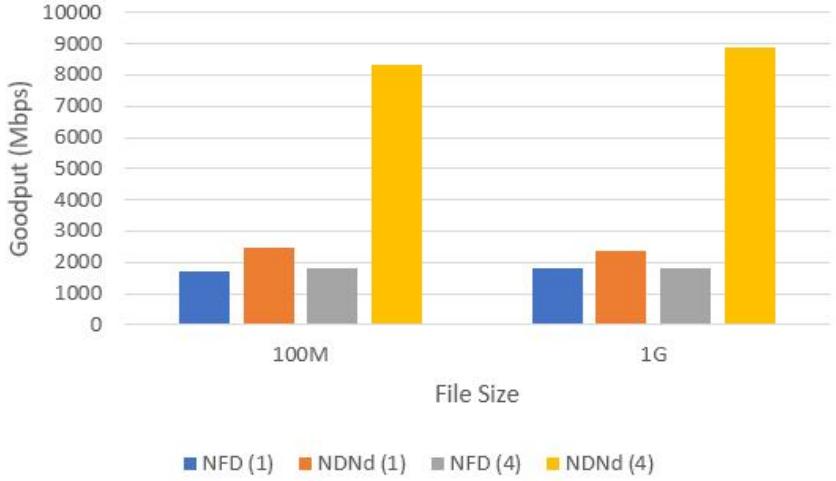
- Sets up a 52-node topology
- Runs simple experiments that need routing and forwarding
- Everything runs in CI
  - GitHub Actions

Continuous Benchmarking

https://named-data.github.io/ndnd/dev/bench/index.html

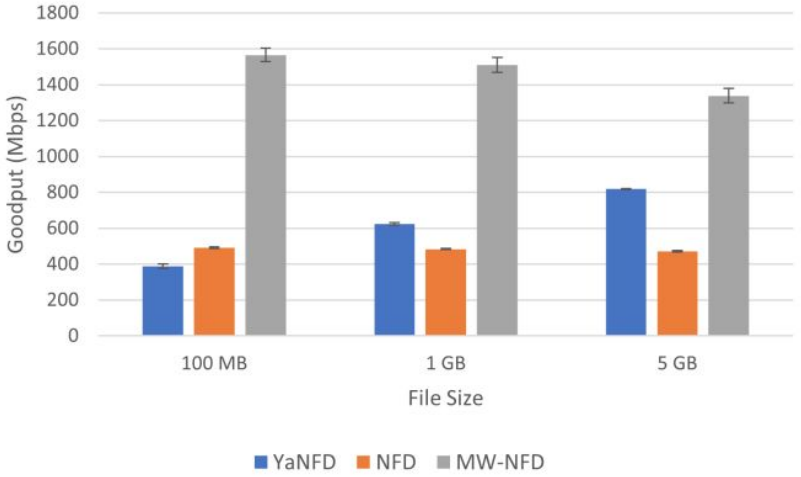


Forwarder Performance



Current Measurements

NDNd w/ 8 fw-threads



Results from 2021 YaNFD paper

YaNFD: Yet another Named Data Networking Forwarding Daemon
ICN '21, September 22–24, 2021, Paris, France

Questions?

https://github.com/named-data/ndnd

Named Data Networking Daemon

build passing | test passing | go report A+ | benchmark graph

release v1.5.0 | license MIT | reference

NDNd is a Golang implementation of the Named Data Networking (NDN) protocol stack.

See the project overview, architecture details and the tutorial for more info on NDN.