# NFD Development Progress

**Beichuan Zhang**

**The University Of Arizona**

# NFD: NDN Forwarding Daemon

**A year ago, we made the first public release**

- Open source (GPL3+)
- New flexible packet format based on TLV
- Modular and extensible design
- Support for multiple forwarding strategies

# Progress in the past year

**Completed the transition from CCNx/NDNx to NFD**

- All software in NDN project is now based on ndn-cxx, NDN-CPP, jNDN, NDN-JS, PyNDN libraries.

**One major release and five minor releases. Another major release (0.4) coming soon**

- beta today, and full release next month.

**We've been expanding supported platforms, adding new features and fixing bugs.**

**Maintain the model of open source development and community contributions.**

# Active Development

## Weekly code commits at Github

### NFD


### ndn-cxx library

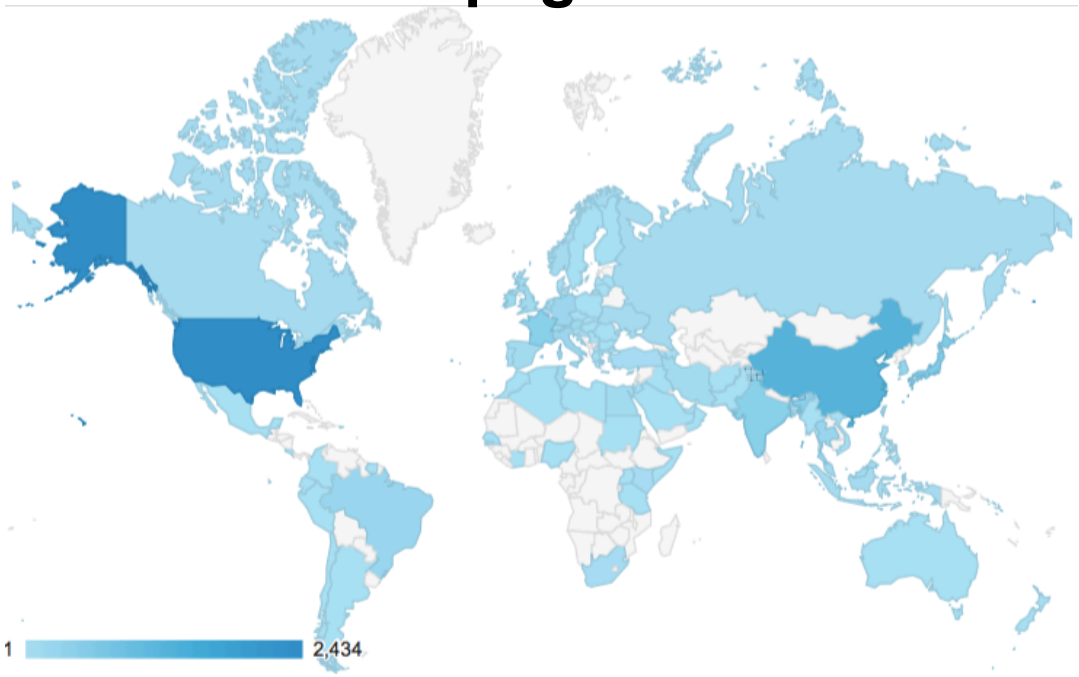
# Community Involvement

**nfd-dev mailing list members: ~100**

**Forks of NFD at Github: 21**

**Code contributors: 25 (some outside of NDN team)**

**Access to the NFD web page:**



# NFD now runs on Android!

**Native code compilation**

- https://github.com/named-data-mobile/NFD-android

**Preliminary tests on performance and energy**

**A few pilot applications**

- Simple game
  - https://github.com/dchimeraan/ndn-hangman
- NDN Whiteboard
  - https://github.com/sumitgouthaman/NDNWhiteboard
- Photo sharing app
  - https://github.com/ohnonoho/photoSharing

# Some other platforms

**Raspberry Pi**

- Used to prototype smart home devices

**DD-WRT and OpenWrt**

- Home routers

**Other embedded systems**
http://redmine.named-data.net/projects/ndn-embedded/wiki

# Simulator Integration

**ndnSIM is a widely used simulator**

- ~185 citations
- ~300 members in the mailing list

**ndnSIM 2.1 uses most NFD code for the forwarding.**

- Easy transition from simulation code to real systems
- Simulation results closer to that from real systems.
- Allows simulation using real application code

http://ndnsim.net/2.1/

# mini-NDN

**A light-weight emulation tool based on mininet**

- Virtualized nodes that run NFD, NLSR, and other NDN applications.
- Easy configuration of topology and link properties such as delay, bandwidth and loss.
- A physical server can run an NDN network of tens of nodes.
- Successfully used in evaluating routing protocols.

https://github.com/named-data/mini-ndn

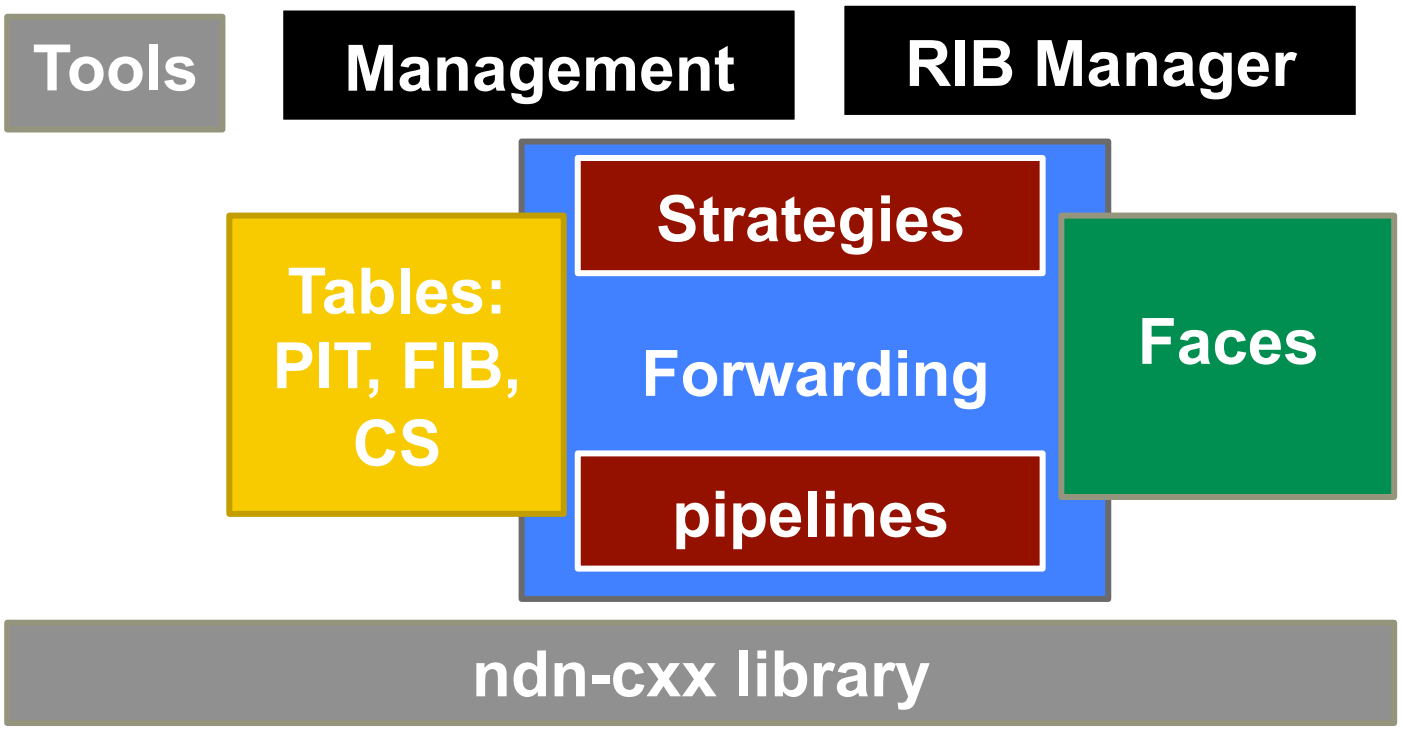
# Evaluation Platforms

**Every release of NFD is tested and deployed on the global NDN testbed.**

**For evaluation, users now have a set of choices with different tradeoffs between scale and fidelity.**

- ndnSIM
- Mini-NDN
- Open Network Lab
- NDN testbed

# Major Components of NFD



# LINK Object

**LINK is a new type of content object, which links one name to another.**

| Name (/net/ndnsim/LINK) |
| --- |
| ContentType=LINK |
| /att/users/alex/net/ndnsim, 100<br>/ucla/users/alex/net/ndnsim, 50 |
| Signature |

**To support mobility, and routing scalability.**

**http://redmine.named-data.net/issues/2587 TR coming soon**

# Producer NACK

**Another new type of content object, to notify consumers that the content doesn't exist yet.**

| Name |
| --- |
| ContentType=NACK |
| • Name (prefix) of non-existent content<br>• A code of why it's not available<br>• Expiration time of this NACK |
| Signature |

# Network NACK

**When a node cannot fetch the data, generate a NACK to signal the downstream to explore other options.**

- Loop/duplicate, link failure, no route, congestion, …

**Return the unsatisfied Interest together with an error code as the NACK**

**Downstream node explores other forwarding options.**

http://redmine.named-data.net/projects/nfd/wiki/NDNLPv2

http://redmine.named-data.net/issues/2930

# NDNLPv2

**Within one hop, under the NDN Interest/Data layer.**

**A set of link services over underlying transport**

- Fragmentation/reassembly
- Loss detection/recovery
- Link failure detection
- Network NACK

**Services are optional depending on the type of transport**

- E.g., TCP, UDP, Ethernet

# NDNLPv2

**Also used between local apps and NFD for control, management and monitoring**

- Specify nexthop for Interests
- Learn incoming face of packets
- Whether to cache an outgoing Data or not

http://redmine.named-data.net/projects/nfd/wiki/NDNLPv2

# Forwarding Strategy

**Version 4 of the Best Route Strategy**

- Support Interest retransmission with exponential backoff of the suppression interval.
- Support LINK object for mobility and routing scalability.

**The Access Strategy for end hosts**

- Multicast to learn which host provides the content and remember what has been learned.

**The Adaptive SRTT-based Forwarding strategy for hyperbolic routing.**

# Management

**An important part of NFD, an interesting app in its own.**

- Process control commands: parsing, verification, dispatch, execution.
- Publish results: StatusDataset or NotificationStream.

**Completely refactored**

- APIs for other NDN programs to reuse the same mechanisms.
- New features and improvements
  - Publish routable prefixes as part of autoconfig
  - Support face query.

# Face System

**Refactored to support NDNLPv2**

- Transport part to send/receive NDNLP packets
- LinkService part to make proper Interest/Data packets

**Add support for “permanent” face**

- Automatically recover from underlying socket errors.

# Content Store

API for customized cache replacement policy

Support no-cache option from local apps.

Compute digest only when needed.

# NDN Essential tools

- **ndnpeek/ndnpoke**: transmit a single packet between a consumer and a producer
- **ndnping/ndnpingserver**: test reachability between two nodes
- **ndndump**: analyze traffic on the wire
- **ndn-dissect**: inspect TLV structure of NDN packets
- **ndn-dissect-wireshark**: Wireshark extension to inspect TLV structure of NDN packets
- **ndn-pib (PIB)**: a service to manage the public keys and publish certificates

# Routing

**NLSR**

- Security implemented and deployed on testbed
- Performance and usability improvements

**Hyperbolic routing**

- Using coordinates instead of building paths.
- Design, implementation and performance evaluation.

**SNAMP: realize the idea of map-and-encap to scale global routing**

- Make use of LINK object and NDNS service.

# Security

PIB service to manage public keys and publish certs

New NDN certificate format

Support the ECDSA signature

Improved signing APIs for better usability

Automated testbed certificate issuance

Applying the name-based security model to NLSR and NFD management.

# Edge Support

**Autoconfig and local hub discovery**

- Combination of various techniques to automatically discover and connect hosts to NDN testbed.
- http://named-data.net/doc/NFD/current/manpages/ndn-autoconfig.html
- http://named-data.net/doc/NFD/current/misc/local-prefix-discovery.html

**Automatic Prefix Propagation**

- Producer connects to gateway and securely register its content prefixes with the gateway.
- Needed for the last hop delivery of interests to the producer.
- TR coming soon

# Future Plan

**Forwarding Strategy**

- New strategies to support IoT, sensors, mobile and DTN environments.
- Composable strategy towards the vision of a limited VM.

**NDN over constrained communication channels**

**Scoped communication**

- within enterprise, homes, etc.

**Hop-by-hop interest limit mechanism for congestion control**

**Packet format refinements**

**Performance optimizations**

- packet processing, data structures and algorithms, crypto.