Overview of NFD

Beichuan Zhang

The University Of Arizona

What is NFD

A new NDN Forwarder, implementing the NDN protocol.

Why NFD

To support the new TLV packet format

To facilitate research and experimentation

- Modularity and extensibility
- Allow easy experimentation with new protocol features, algorithms, and applications
- Reasonable performance

To provide free, open-source NDN implementation for the community

Development Platform

Base

- C++, waf build framework, Boost library

In-project library

- ndn-cxx

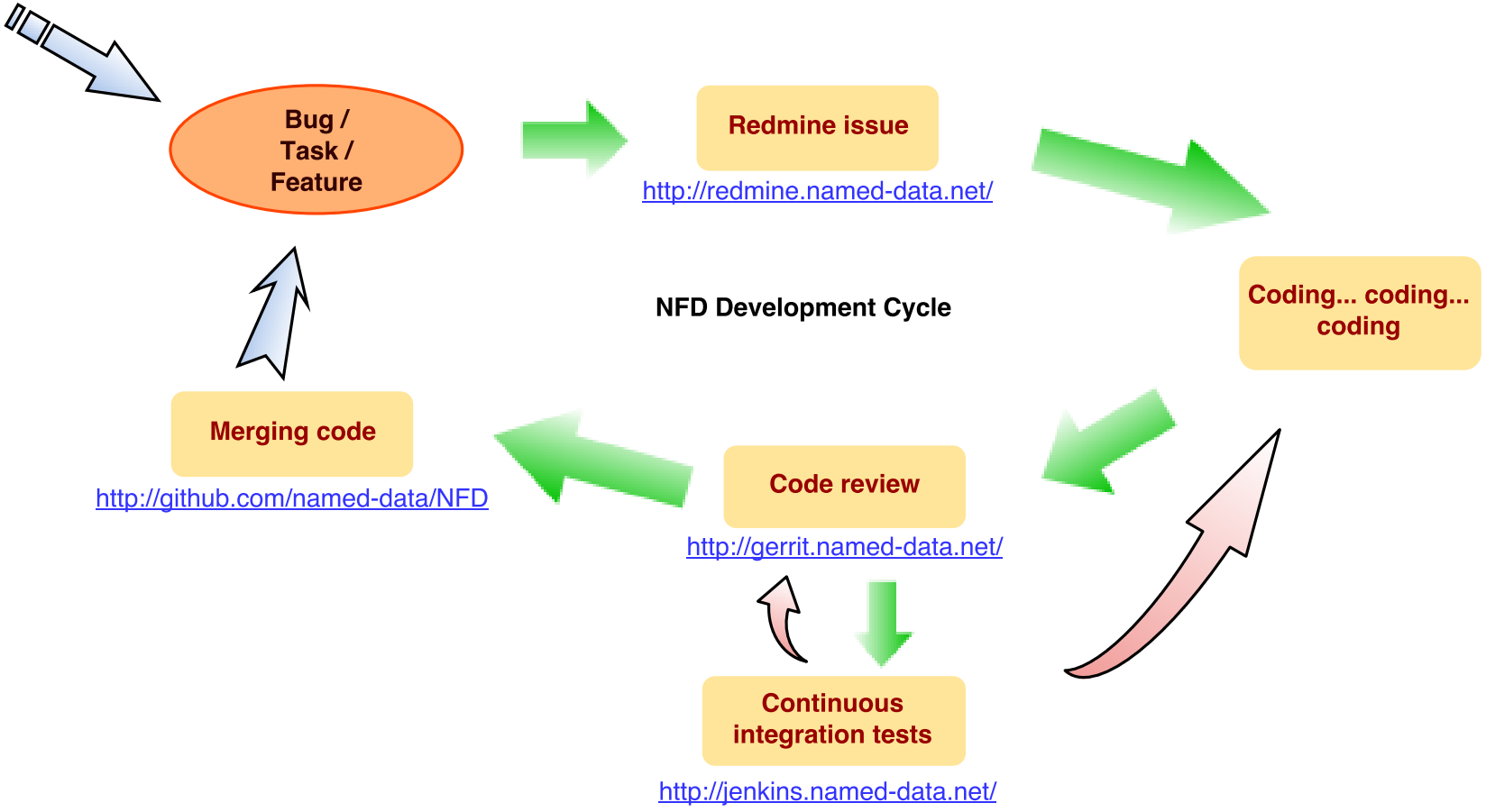
Documentation

- Doxygen for API documentation
- Sphinx for overall documentation
- Redmine Wiki for volatile docs and user-generated documentation

Development Model

Continuous Integration:

- Redmine, Jenkins, Travis-CI



Who did it

Credits to all NFD developers

UCLA:

- Alexander Afanasyev, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Lixia Zhang

Arizona:

- Junxiao Shi, Yi Huang, Jerald Abraham, Beichuan Zhang

Colorado State:

- Steve DiBenedetto, Chengyu Fan, Christos Papadopoulos

WashU:

- Haowei Yuan, Hila Ben Abraham, Patrick Crowley,

Memphis:

- Syed Obaid Amin, Vince Lehman, Lan Wang

UPMC:

- Davide Pesavento, Giulio Grassi, Giovanni Pau,

BIT:

- Hang Zhang, Tian Song,

Release Schedule

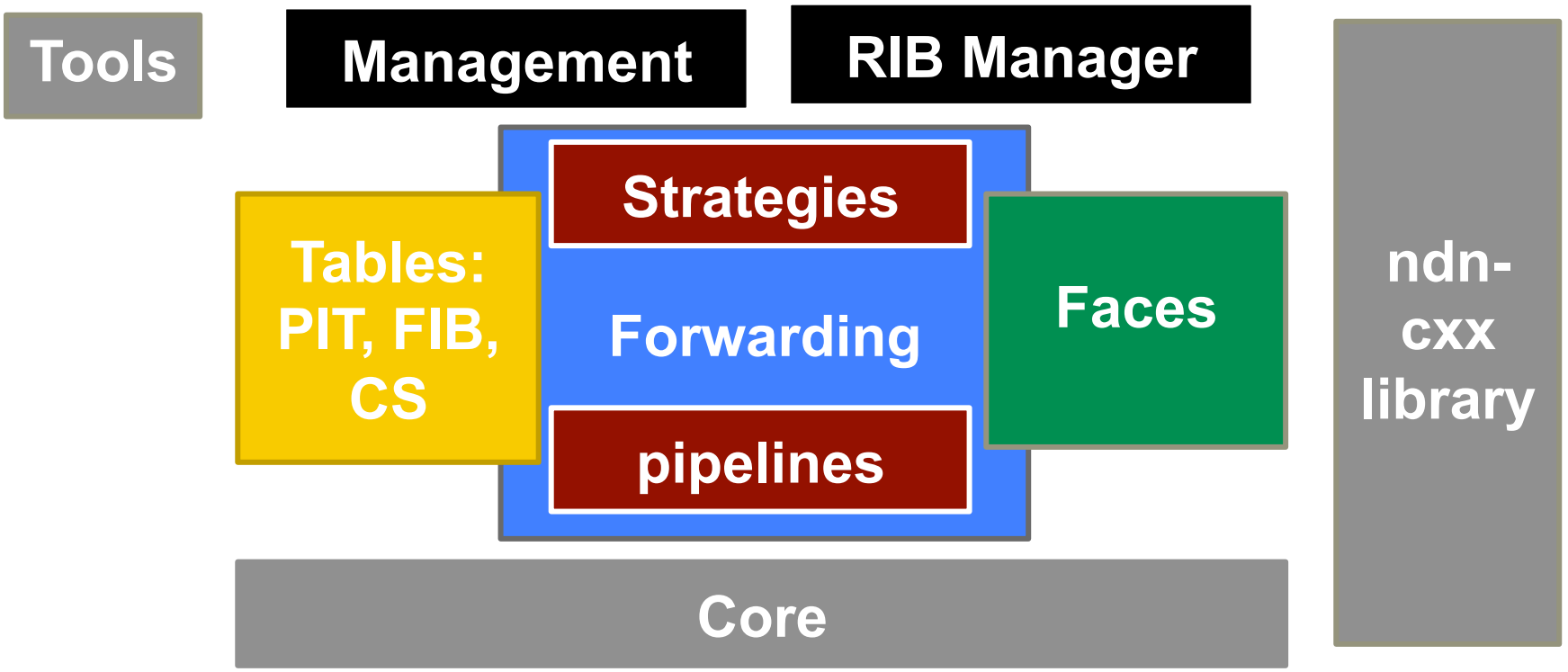
Initial public release (version 0.2.0)

- On August 25th, 2014
- Extensive documentation set, including NFD Developer's Guide.
- Target platforms: OSX, Ubuntu
- Known to work on: RedHat, Gentoo, FreeBSD, Raspberry Pi, OpenWRT

Future releases

- ~ Every 3 months
- Evolve together with the protocol spec.
- New features, new platforms
- Performance improvements

Major Pieces



Core

Hash computation routines

- city hash

Basic logger

- multiple log levels
- log level can be configured for individual module

Configuration file

- Boost INFO format

DNS resolver

- async and sync resolution helpers

Tables

In the initial release, mostly follow the CCNx 0.7 design

- To ensure all functionality is there
- But make it easier to adopt new designs in the future
- Name Prefix Hash Table (NPHT) for combined PIT and FIB, plus new Measurement and StrategyChoice tables.
- SkipList for CS implementation

Notable changes

- Distinct **Measurements** table to keep track strategy-specific measurement information
- **StrategyChoice** table to record per-namespace strategy choices
- PIT and Measurement entries can store strategy-related information
- FIB is separated from RIB, and RIB management moved to a separate process

Faces

Face abstraction

- Receive and send Interest and Data packets
- Unicast vs. multicast
- Local vs. remote

Initial release:

- TCP, UDP, Unix socket, Ethernet, Websocket

Forwarding

Packet processing is broken into a number of small "pipelines"

- E.g., incoming Interest, Interest loop, outgoing Interest, Interest reject, Interest unsatisfied, etc.

A strategy provides a set of callback functions at different stages of packet processing

- per namespace, local to the node, extended PIT/FIB structures, new StrategyChoice and Measurements tables.
- can be configured via management commands
- Initial release: best route, broadcast, client control, ccnx.

Management

NFD provides APIs for authorized programs to configure and manage the forwarder

- Interest/Data exchange

Face management

- creation, destroy

Prefix and strategy management

- Prefix registration
- FIB add/remove/modify
- Strategy selection

Status and statistics retrieval

Tools

nfd-start/nfd-stop

- Helpers to start/stop the forwarder

nfdc

- change parameters at run time via management interface

nfd-status, nfd-status-http-server

- Read status and statistics via management interface, serve it as simple HTML page over HTTP

Auto configuration

- Use DHCP and DNS to find a local or remote gateway

More information

http://named-data.net/doc/NFD/current/

Tutorial at ICN conference

Feedbacks, suggestions, and contributions are welcome.