

Enhanced Testbed Monitoring and FCH Service

Alexander Lane
Davide Pesavento

Planned Tasks

1. Develop a service that periodically:
 - connects to a testbed router over UDP, TCP, and WSS,
 - sends ndnping probes to every known destination,
 - registers a prefix and checks its propagation by pinging from another router.

Store collected results into a database.

2. Develop a web application that shows the results from the database.
3. Develop an NDN-FCH compatible server that responds with routers with minimum downtime in the past T hours. This server can directly connect to the database.
4. Improve ndn-autoconfig client: try multiple routers from NDN-FCH, try the Wi-Fi access point, etc.

It was supposed to be a 4-person project...

What We Accomplished

- Pinger program, written in Python3 and PyNDN
- Fetch master list of testbed hubs (JSON file)
- “Connect” to each hub over TCP, UDP, WSS, one protocol at a time
- Register a prefix
- Ping that prefix by sending Interests from all other hubs
- Collect and store results in SQLite database

```
davide@ndnbox2:~/testbed-status$ PYTHONPATH=./PyNDN2/python python3 pinger.py
Adding faces to hub: University oh Minho
Begin ping
Begin ping
Begin ping
Adding faces to hub: University of Basel
Begin ping
Begin ping
Begin ping
Adding faces to hub: Austrian Alpen-Adria-Universitat
Begin ping
Begin ping
Begin ping
Adding faces to hub: University Michigan
Begin ping
Begin ping
Begin ping
Adding faces to hub: University of Padua
Begin ping
Begin ping
Begin ping
Adding faces to hub: University Pierre et Marie Curie, Sorbonne Universities LIP6
Begin ping
Begin ping
Begin ping
Adding faces to hub: Anyang University
Begin ping
Begin ping
```

Prefix Readvertisement Woes

- Send control command to `/localhop/nfd/rib/register`
- “Origin” field of control parameters **MUST** contain a magic number (65) for client-to-nlsr readvertisement to work
 - Not documented anywhere?
- Even with the right Origin value, the control command was still being rejected by the remote NFD
 - Impossible to debug
 - RIB management protocol does not document potential error codes, or anything else regarding the format of error messages
 - PyNDN does not provide a reason in the prefix registration failure callback
 - NFD knows the root cause (in this case a slightly out-of-sync clock) but only encodes a generic “command rejected” in the response packet

Prefix Readvertisement Woes

- Debugging
 - SSH access to testbed router (LIP6)
 - Enable debug logging (which module? Hint: not what you'd think)
 - Command interest validator is in ndn-cxx (how to discover logger module name?)
 - Need to group logging by logical function performed, not by C++ class that implements it
 - *“Timestamp is outside the grace period for key /foo/bar/...”*