CCN-lite

Nikos Fotiou
fotiou@aueb.gr
CCN-lite (http://www.ccn-lite.net/)

• Small CCNx-compliant core
  – Less than 1000 lines of code

• Runs in
  – Unix user space
  – Linux kernel
  – Omnet++
  – X86 and ARM
  – RIoT
What CCN-lite supports*

- the classic CCNx ccnb version (Nov 2013)
- the Named-Data-Networking (NDN) protocol (as of Nov 2013)
- the CCNx1.0 encoding (unofficial, as of Dec 2014)
- an experimental and compact encoding for IoT environments (Nov 2014)
- the novel Named-Function-networking approach (University of Basel)

* According to its documentation
What CCN-lite supports*

- ccnb encoding of messages (or any other encoding flavor, including the TLV variants)
- PIT and FIB (CCN basic data structures)
- Longest and exact prefix matching (for basic CCN operations)
- Matching of publisher's public key (to fight cache poisoning)
- Nonce and/or hop limit tracking (to avoid loops as a minimal safeguard)
- Packet fragmentation

* According to its documentation
What CCN-lite does not support*

- sophisticated data structures for performance optimizations
- exclusion filtering in interests
- all TCP connectivity and the old CCNx cmd line utilities that have TCP hardwired into them
- crypto functionality, which here is not our prime concern
- repository functionality, SYNC server etc

* According to its documentation
Installation

• git clone https://github.com/cn-uofbasel/ccn-lite
• Make all
Create Content

$ ccn-lite-mkC -s ndn2013 -o ~/Contents/mycontent.ndntlv “/ndn/test/mycontent”
Create Content

$ ccn-lite-mkC -s ndn2013 -o ~/Contents/mycontent.ndntlv "/ndn/test/mycontent"

• Command for creating content
Create Content

$ ccn-lite-mkC -s ndn2013 -o ~/Contents/mycontent.ndntlv "~/ndn/test/mycontent"

• Command for creating content
• Wire format
  – ccnb, ccnx2014, ndn2013
Create Content

$ ccn-lite-mkC -s ndn2013 -o ~/Contents/mycontent.ndntlv "/ndn/test/mycontent"

• Command for creating content
• Wire format
• Output file
Create Content

```
$ ccn-lite-mkC -s ndn2013 -o ~/Contents/mycontent.ndntlv "~/ndn/test/mycontent"
```

- Command for creating content
- Wire format
- Output file
- Name of the content
Create Content

• Then type the contents in a single line
• Otherwise use –i <INPUT FILE NAME>

• Be careful:
  – Output filename must have the correct extension
    • ndntlv, ccntlv, ccnb
  – Output filename must be the suffix of the content name
    • mycontent.ndntlv “/ndn/test/mycontent”
Create content router (ccn-lite-relay)

ccn-lite-relay -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents
Create content router (ccn-lite-relay)

**ccn-lite-relay** -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents

- Command for creating the content router
Create content router (ccn-lite-relay)

```
ccn-lite-relay -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents
```

- Command for creating the content router
- Wire format
Create content router (ccn-lite-relay)

ccn-lite-relay -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents

- Command for creating the content router
- Wire format
- Port to listen
Create content router (ccn-lite-relay)

ccn-lite-relay -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents

• Command for creating the content router
• Wire format
• Port to listen
• Socket used for sending management commands
Create content router (ccn-lite-relay)

• ccn-lite-relay -s ndn2013 -u 9999 -x /tmp/a.sock -d ~/Contents

• Command for creating the content router
• Wire format
• Port to listen
• Socket used for sending management commands
• Directory where content items are stored (if any)
Send Interest

ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "/ndn/test/mycontent"
Send Interest

```
ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "/ndn/test/mycontent"
```

• Command for creating the interest
Send Interest

`ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "/ndn/test/mycontent"`

- Command for creating the interest
- Wire format
Send Interest

ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "/ndn/test/mycontent"

• Command for creating the interest
• Wire format
• Where to send the interest
Send Interest

ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "/ndn/test/mycontent"

- Command for creating the interest
- Wire format
- Where to send the interest
- The name of the content item
Interpreting response

- ccn-lite-pktdump parses rackets
- A response can be pipelined to ccn-lite-pktdump e.g.,

```
ccn-lite-peek -s ndn2013 -u 127.0.0.1/9999 "'/ndn/test/mycontent"| ccn-lite-pktdump
```
Adding a second content router

ccn-lite-relay -s ndn2013 -u 9998 -x /tmp/b.sock
Create iface and get ifaceid

ccn-lite-ctrl -x /tmp/b.sock newUDPface any 127.0.0.1 9998 | ccn-lite-ccnb2xml | grep FACEID
Create iface and get ifaceid

ccn-lite-ctrl -x /tmp/b.sock newUDPface any 127.0.0.1 9998 | ccn-lite-ccnb2xml | grep FACEID

• The socket used for administration
Create iface and get ifaceid

ccn-lite-ctrl -x /tmp/b.sock newUDPface any 127.0.0.1 9998 | ccn-lite-ccnb2xml | grep FACEID

• The socket used for administration
• Source and destination
Create iface and get ifaceid

ccn-lite-ctrl -x /tmp/b.sock newUDPface any 127.0.0.1 9998 | ccn-lite-ccnb2xml | grep FACEID

• The socket used for administration
• Source and destination
• Decode ccn file
Add forwarding rule

ccn-lite-ctrl -x /tmp/b.sock prefixreg /ndn <FACEID> ndn2013 | ccn-lite-ccnb2xml