Introduction to CCN-lite

Christopher Scherb, Claudio Marxer, Christian Tschudin

University of Basel
Department for Mathematics and Computer Science
Computer Networking Group

ACM ICN 2017

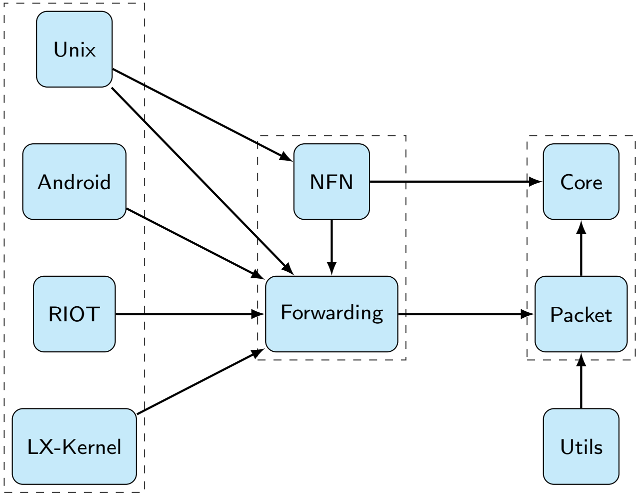
Introduction

- CCN-lite is a lightweight ICN implementation
- permissive ISC license
- developed at University of Basel
- Multi Packet format forwarder: NDN, CCNx, etc.
- CCN-lite runs on multiple platforms
    - x86/64 on Linux, BSD and MacOS, Kernel Module for Linux
    - Android, Arduino
    - ARM Cortex A-series
    - RIOT (e.g. ARM Cortex M-series)

CCN-lite v2

- CCN-lite was started 2011
- complete restructure of the Code (2017)
- split the code in several modules
- available as libs and source modules
- modules can be disabled at compile time
- provide packet encoding library for applications
- remove code duplications

Structure of CCN-lite v2



CCN-lite Home structures

- CCN-Lite Home Dir
  - doc
  - src
    - ccnl-core
    - ccnl-fwd
    - ccnl-pkt
    - ccnl-unix
    - ccnl-riot
    - ccnl-utils
    - ...
  - test
  - tutorial

CCN-lite Usage: Build Process

Universität Basel

- Source Code available on Github
  `github.com/cn-uofbasel/ccn-lite`
- CMake Build System
- Dependencies: OpenSSL, CMake
- only two dependencies, fast compiling ⇒ easy to start
- Building CCN-Lite from CCN-Lite Home Dir:

```
mkdir build
cmake ../src
make
```

CCN-lite Command Line Tools

Selection of Command Line Tools important for this tutorial

`ccn-lite-relay`	Unix forwarder
`ccn-lite-ctrl`	Tool for configuration
`ccn-lite-ccnb2xml`	Print CTRL packets
`ccn-lite-mkC`	Tool to create data objects
`ccn-lite-mkI`	Tool to create interest packets
`ccn-lite-peek`	Tool to fetch a data object
`ccn-lite-pktdump`	Tool to analyze packets

ccn-lite-relay 1

- Unix forwarder for ICN
- Supports Linklayer, 802.15.4, UDP, Unix-Socket communication
- single binary: forwarding of different packet formats
- Supports NFN forwarding layer (Named Function Networking)

ccn-lite-relay 2

```
-d databasedir
-e ethdev
-s SUITE        (ccnb, ccnx2015, cisco2015,
                    iot2014, ndn2013)
-w wpandev
-u udpport      (can be specified twice)
-6 udp6port     (can be specified twice)
-v DEBUG_LEVEL  (fatal, error, warning, info,
                    debug, verbose, trace)
-x unixpath
```

```
ccn-lite-relay -v debug -x /tmp/mgmt.sock -u 9000
```

ccn-lite-ctrl 1

Management system for the ccn-lite-relay

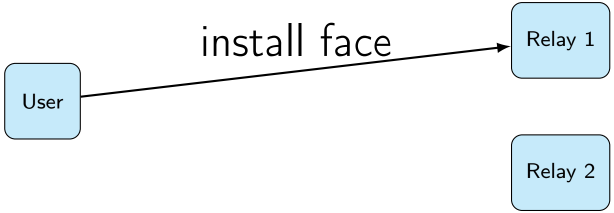
```
ccn-lite-ctrl
    [-h] [-k relay-public-key] [-m] [-p private-key]
    [-v debug level]
    [-u ip-address/port | -x ux_path]

    CMD
```

ccn-lite-ctrl 2: commands

```
newETHface        MACSRC|any MACDST ETHTYPE
newUDPface        IP4SRC|any IP4DST PORT
newWPANface       WPAN_ADDR WPAN_PANID
newUDP6face       IP6SRC|any IP6DST PORT
newWPANface       WPAN_ADDR WPAN_PANID
newUNIXface       PATH
destroyface       FACEID
prefixreg         PREFIX FACEID [SUITE]
prefixunreg       PREFIX FACEID [SUITE]
addContentToCache             ccn-file
removeContentFromCache        ccn-path
```

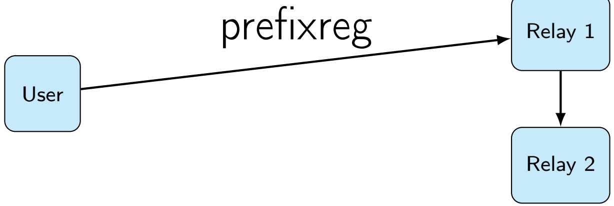
CCN-lite: install a face



```
ccn-lite-ctrl -x /tmp/mgmt1.sock newUDPface
any ip/port | ccn-lite-ccnb2xml
```

creates a new abstract interface on *Relay 1* pointing to another relay (*Relay 2*)

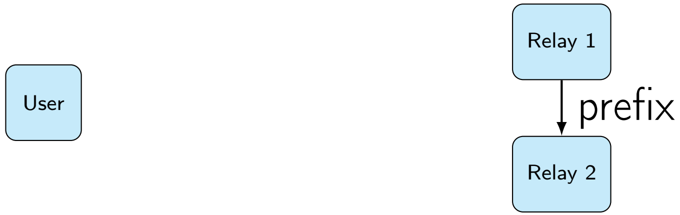
CCN-lite: register a prefix



```
ccn-lite-ctrl -x /tmp/mgmt1.sock prefixreg
prefix faceid | ccn-lite-ccnb2xml
```

creates a new entry in the FIB

CCN-lite Demo Scenario

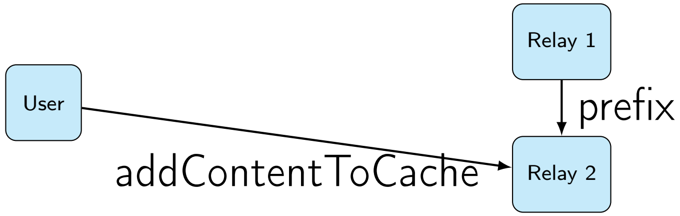


ccn-lite-mkC: create a data object

```
-i FNAME        input file (instead of stdin)
-k FNAME        HMAC256 key (base64 encoded)
-l LASTCHUNKNUM number of last chunk
-n CHUNKNUM     chunknum
-o FNAME        output file (instead of stdout)
-p DIGEST       publisher fingerprint
-s SUITE        (ccnb, ccnx2015, cisco2015,
                 iot2014, ndn2013)
ICN-FILE NAME
```

```
echo "Hello ACM ICN" | ccn-lite-mkC -o mydata.ndntlv
<prefix>
```

ccn-lite-ctrl: addContentToCache



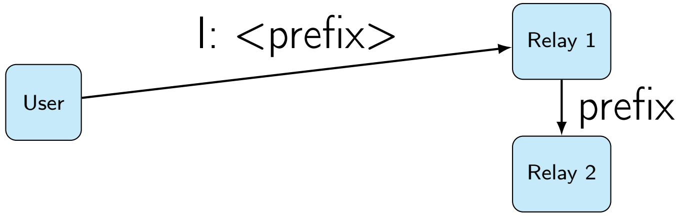
```
ccn-lite-ctrl -x /tmp/mgmt2.sock
addContentToCache mydata.ndntlv
```

ccn-lite-peek 1

```
-n CHUNKNUM       positive integer for
                   chunk interest
-s SUITE          (ccnb, ccnx2015, cisco2015,
                   iot2014, ndn2013)
-u a.b.c.d/port   UDP destination
-v DEBUG_LEVEL    (fatal, error, warning, info,
                    debug, verbose, trace)
-w timeout        in sec (float)
-x ux_path_name   UNIX IPC

ICN-URI
```

ccn-lite-peek 2: fetch data



```
ccn-lite-peek -u ip/port <prefix>
```

CCN-lite Development

- using the packet library for an application
- required libraries: core, pkt
- use header files from src tree
- use libraries from bin/lib tree
- to create an interest:
    - include: "src/ccnl-pkt/ccnl-pkt-builder.h"
    - link with "build/bin/lib/ccnl-core.a, ccnl-pkt.a"

CCN-lite Development: Create an Interest

```
struct ccnl_prefix_s *prefix =
    ccnl_URItoPrefix(char* uri, int suite,
    char *nfnexpr, unsigned int *chunknum)
int nonce = random( )

struct ccnl_interest_s *interest =
    ccnl_mkInterestObject(struct
    ccnl_prefix_s *name, int *nonce)
```

The End

Thank your for your attention!