UCLA ENGINEERING
Henry Samueli School of Engineering and Applied Science
Birthplace of the Internet

NDN ON RIOT-OS

{ WENTAO SHANG, ALEX AFANASYEV, AND LIXIA ZHANG } UCLA

NAMED DATA NETWORKING
RIOT

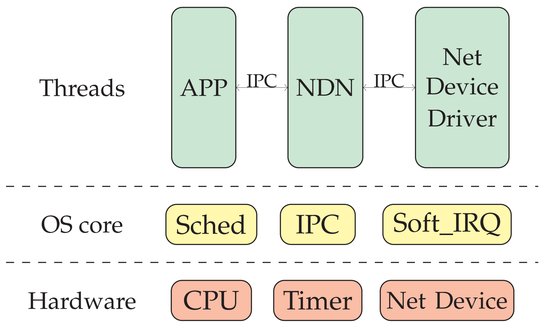
MOTIVATION

- Named Data Networking (NDN) has shown great potential in supporting network applications in the IoT environments [1].
- The goal of this project is to bring NDN protocol support to the constrained IoT devices with 100s of KB memory and low-power CPU.
- We build on top of a popular IoT software platform called RIOT-OS [2].

RIOT-OS FEATURES

- Common OS abstraction across multiple platforms (ARM, Arduino, MSP430)
- Multi-threading + IPC
- Custom network stack
- C/C++ programming environment
- Standard build tools (gcc, make)
- Simulator for testing on Linux PCs

SYSTEM DESIGN



Software architecture of NDN on RIOT-OS

The NDN protocol is implemented as a kernel thread. The IPC channel is used for:

- Passing NDN packets from & to APP and network device driver threads
- Sending configuration commands (e.g., add faces, register prefixes)

Currently implemented features:

- Basic packet forwarding logic (PIT, FIB, CS)
- Support for Ethernet and 802.15.4
- Memory efficient packet encoding & decoding
- HMAC-SHA256 data signing and verification

APPLICATION INTERFACE

The NDN code on RIOT-OS is C99-compatible.

Object	Interface
Name	`ndn_name_from_uri, ndn_name_append, ndn_name_print, ndn_name_compare_block, ndn_name_get_component_from_block`
Interest	`ndn_interest_create, ndn_interest_get_name, ndn_interest_get_nonce, ndn_interest_get_lifetime`
Data	`ndn_data_create, ndn_data_get_name, ndn_data_get_content, ndn_data_get_metainfo, ndn_data_verify_signature`
APP Handle	`ndn_app_create, ndn_app_run, ndn_app_destroy, ndn_app_schedule, ndn_app_express_interest, ndn_app_register_prefix, ndn_app_put_data`

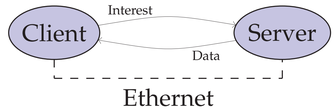
List of API for NDN APP on RIOT-OS

```
static ndn_app_t* handle = NULL;

static int on_data(ndn_block_t* interest, ndn_block_t* data)
{
    ndn_block_t name;
    ndn_data_get_name(data, &name);
    ndn_name_print(&name);
    ndn_block_t content;
    ndn_data_get_content(data, &content);
    // do something with content...
    return NDN_APP_STOP;
}

static int send_interest(void* context)
{
    const char* uri = (const char*)context;
    ndn_shared_block_t* sn = ndn_name_from_uri(uri, strlen(uri));
    ndn_shared_block_t* sin = ndn_name_append_uint16(&sn->block, 0);
    ndn_shared_block_release(sn);
    ndn_app_express_interest(handle, &sin->block, NULL, 1000,
                             on_data, on_timeout);
    ndn_shared_block_release(sin);
    return NDN_APP_CONTINUE;
}

static void run_client(const char* uri)
{
    handle = ndn_app_create();
    ndn_app_schedule(handle, send_interest, (void*)uri, 1000000);
    ndn_app_run(handle);
    ndn_app_destroy(handle);
}
```

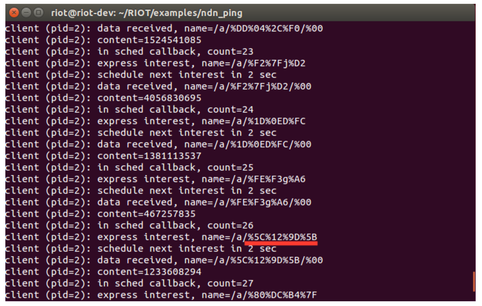
Simple NDN consumer on RIOT-OS

DEMO APPLICATION: NDN-PING

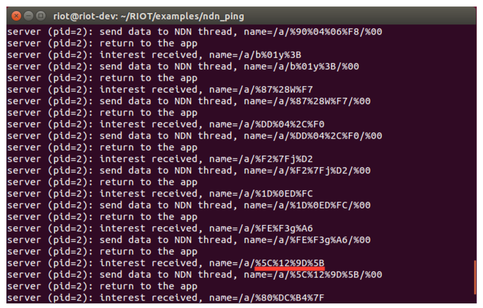
This demo application shows two RIOT-OS nodes running NDN-Ping client and servers respectively in a emulated network environment on a Ubuntu 15.10 machine. NDN packets are sent over Ethernet directly.



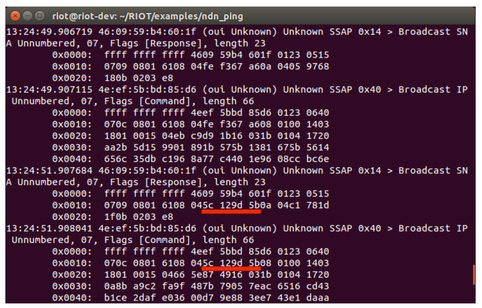
Emulated testbed



NDN-Ping client



NDN-Ping server



Tcpdump output of network packets

text	data	bss	dec	hex	file name
39636	228	11204	51068	c77c	ndn_ping.elf

Code size & static memory usage (compiled for SAM R21 IoT board)

LIMITATIONS & FUTURE WORK

- Currently the code is only tested in emulated environments. The next step is to try it out on a real IoT device.
- The current implementation does not have routing support or FIB/RIB management. An interesting research direction is to provide routing functionality for constrained NDN-IoT networks.
- The current implementation does not include advanced NDN features such as forwarding strategies or cache management policies. It is yet unclear whether it is necessary to support those features on constrained devices.

SOURCE CODE

The source code of this work is available at `https://github.com/wentaoshang/RIOT/tree/ndn/`. It is currently released under LGPL v2.1, the same license used by RIOT-OS itself.

ACKNOWLEDGMENT

This work has been supported by the National Science Foundation under award CNS-1345318, CNS-1345142, CNS-1455794, and CNS-1455850.

REFERENCES

[1] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang. Named Data Networking of Things. In *Proceedings of 1st IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI'2016)*. (Invited paper).

[2] RIOT - The friendly Operating System for the Internet of Things. `http://www.riot-os.org/`.