Named Data Networking of Things: NDN-RIOT, NDN-PI

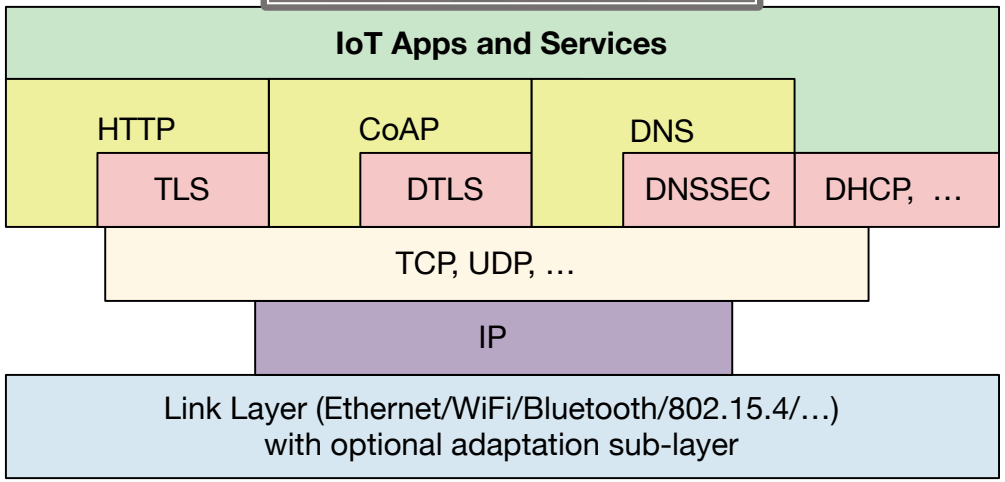
Alex Afanasyev

Florida International University

IoT over ICN Tutorial @ ACM ICN 2017
September 26, 2017
Berlin, Germany

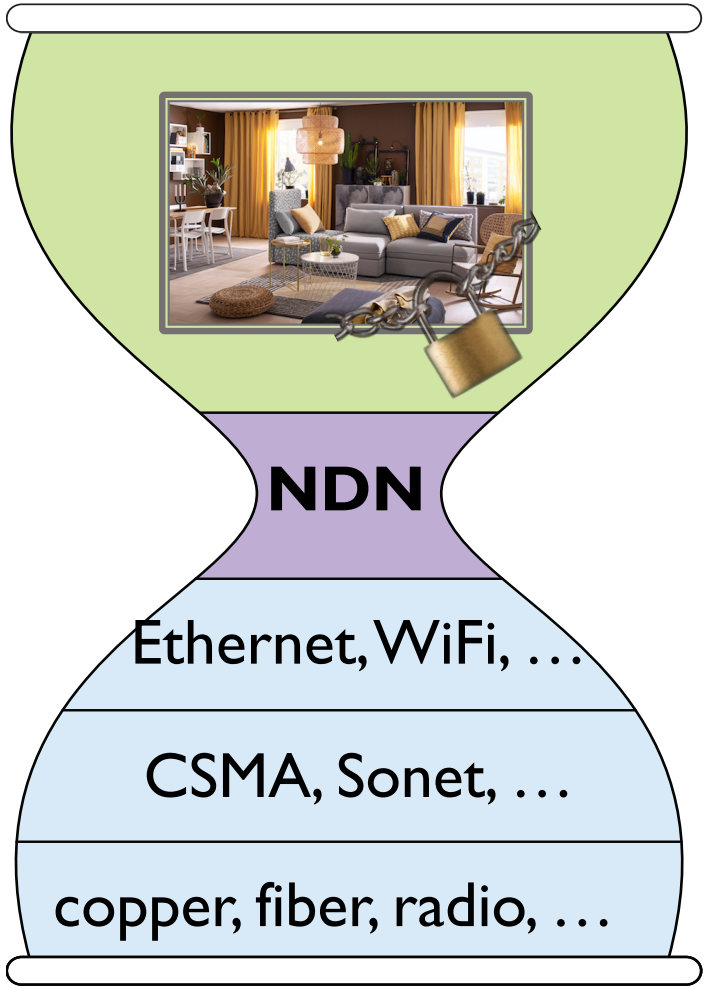
Complexity and Semantic Mismatch for IP/IoT

- **App: "Living room frontal view feed"**
- Network:
  - Request stream (HTTP/CoAP)
  - Connect to camera (TCP/IP)
- +
  - Lookup mapping "Living room" -> camera URI
  - Connect to AlexHome.com (cloud?) service
  - DNS lookup IP of AlexHome.com service
  - DHCP to assign IP addresses to all devices



Named Data Networking of Things

- **App: "Living room frontal view feed"**
  - /AlexHome.com/LivingRoom/VideoFeed/FrontView/mp4/_frame=12/_chunk=20
- Network:
  - Use the name to send request to my camera responsible for Living's room front view
  - OR retrieve data from caches
- +
  - Cameras provision with "identity name" that defines what they are and what data names they produce
  - Can announce name prefixes or respond to local broadcasts



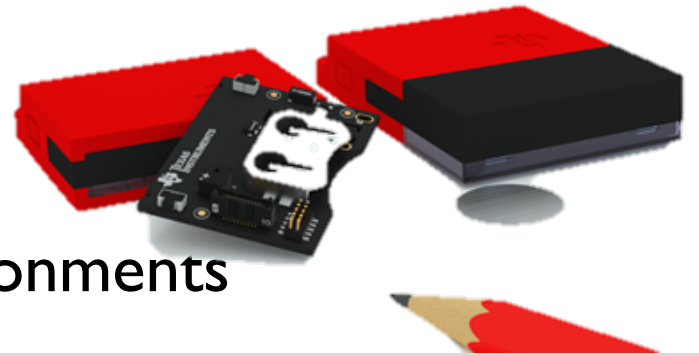
ICN/NDN "Edge" for IoT

- Bring IoT semantics to the network layer
- Name the "things" and operations on "things"
  - "Living room frontal view feed", "CO level in kitchen"
  - "blood pressure", "body temperature"
  - "max/min/avg pH of soil in specific point of US soil grid"
- Focus on data associated with things, not devices
- Secure data directly

W. Shang et. al, "Named Data Networking of Things," in proc. of IoTDI'2016

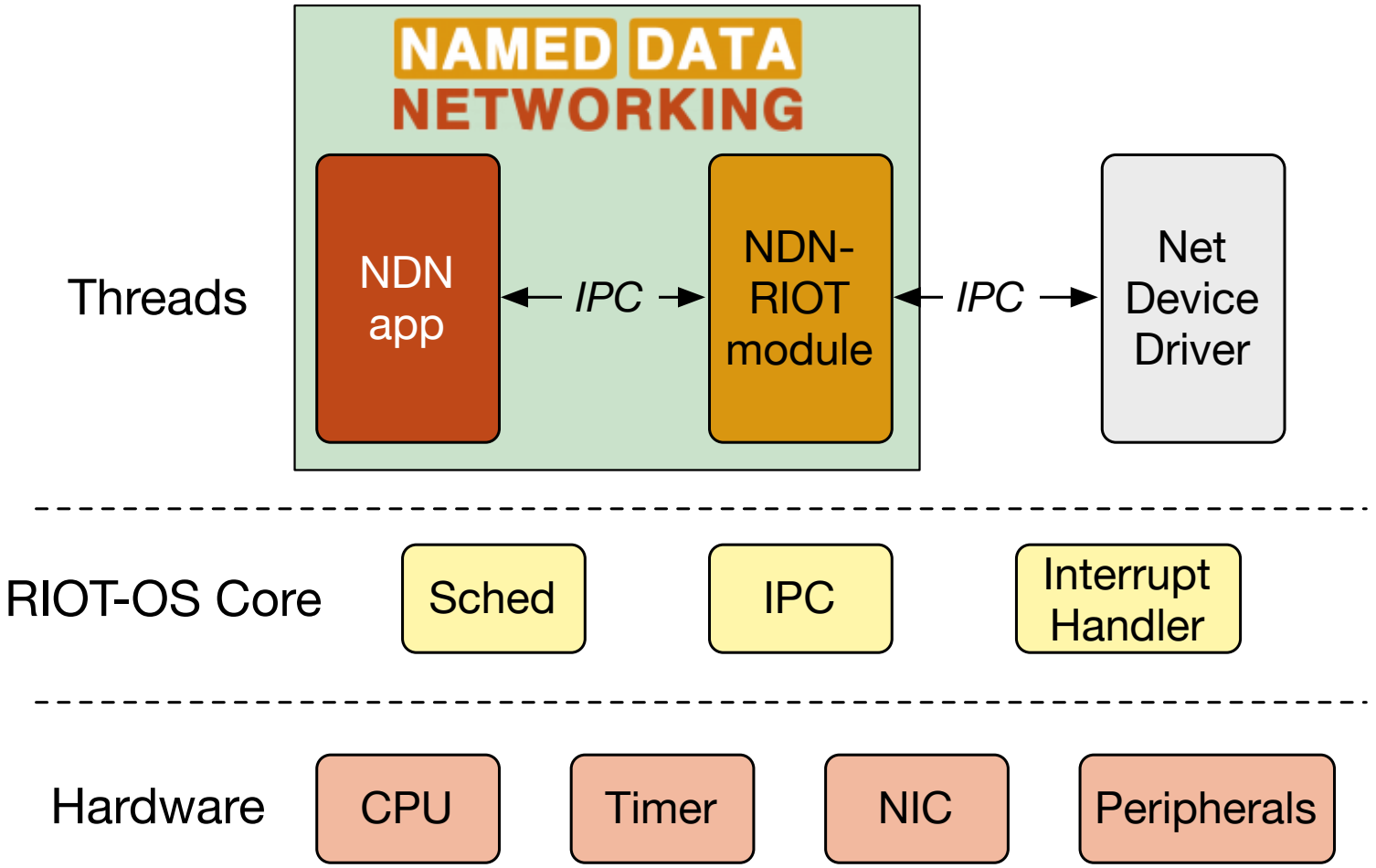
Smarter IoT with Low-cost Devices

- Hardware: ultra low cost, longevity
  - Constrained battery, low-power networking, limited memory, low CPU
  - ~ 32-bit ARM, 48 MHz, 32KB RAM, 256KB flash
- Application getting smarter and more powerful
  - Need integration with public Internet and cloud service without requiring gateways
  - Need for data-centric security, local trust management
  - Need auto-discovery and auto-configuration
- **Named Data Networking**
  - common protocol for all applications and network environments



W. Shang, et. al, "Breaking out of the cloud: local trust management and rendezvous in Named Data Networking of Things," to appear in IoTDI'17

NDN-RIOT Architecture Highlights



- Support for NDN packet format for limited MTU links
- Support of data-centric security, including ECDSA and HMAC signatures, AES encryption
- Replaceable forwarding strategies
- Support of transmission (+fragmentation) over IEEE 802.15.4 and Ethernet
- Simple application API

- A few basic examples
  - https://github.com/named-data-iot/ndn-riot-examples

Open source, contributions welcome

https://github.com/named-data-iot/ndn-riot

Memory-Optimized Packet Decoding

- Shared memory block structure to move packets
  - avoid memory copy in most cases
- On-demand packet field extraction
  - avoid memory for decoded meta data

Security Support

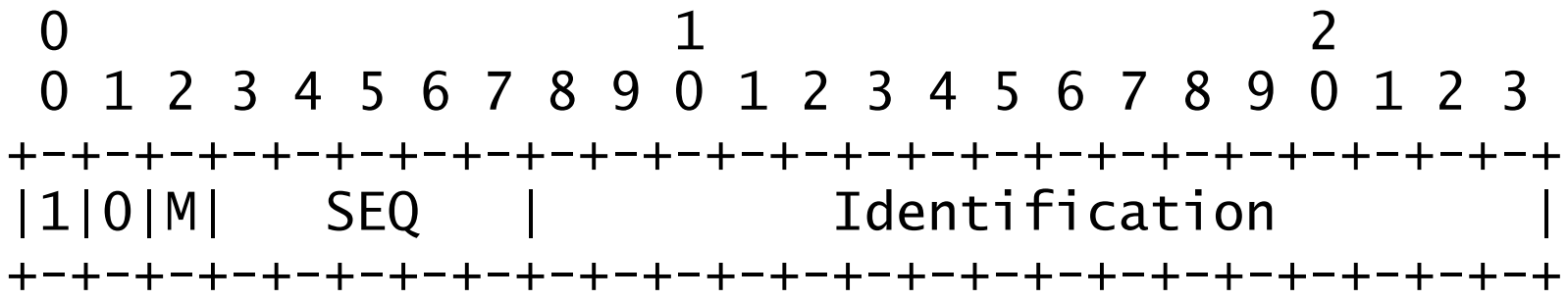
- ECDSA
  - micro-ecc library (https://github.com/kmackay/micro-ecc)
  - secp256r1 curve with 64-byte signatures
  - deterministic signing (RFC 6979) given lack of good entropy on many current devices
    - keys need to be generated outside the device
- no RSA
  - too much overhead and too expensive to produce signatures
- HMAC
  - RIOT-OS built-in APIs

Packet Forwarding

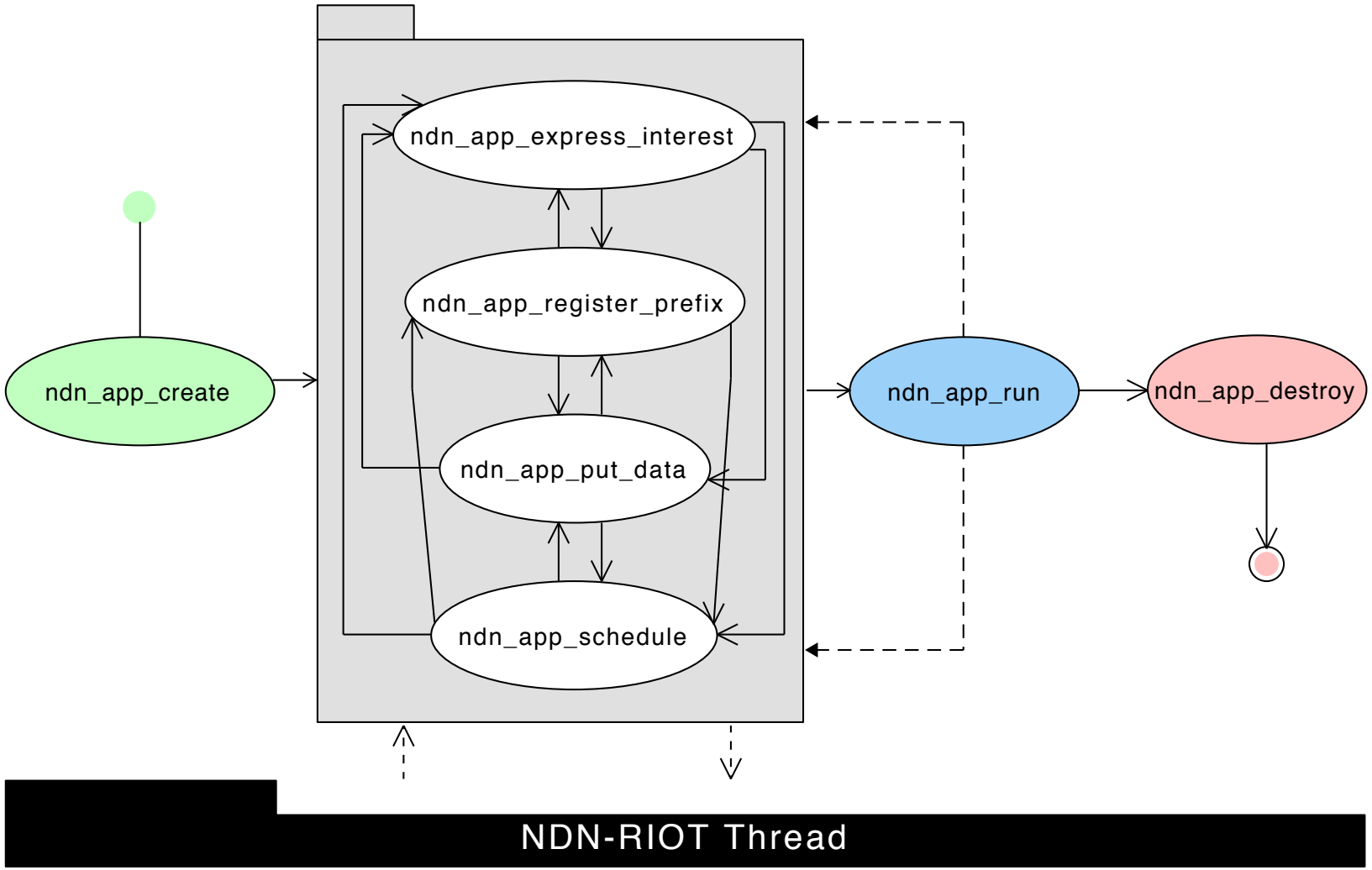
- PIT
  - exact match for interest
  - “any” prefix match for data (all interests that are prefix of the data)
- FIB
  - longest prefix match for interest names
  - static compile-time prefix registration
  - IPC-based run-time prefix registration (for local apps)
- CS
  - “any” match for interests (a data for which interest is a prefix)
  - compile-time adjustable size (~24KB default settings)
  - FIFO policy
- Replaceable forwarding strategies
- Work in progress
  - Support for basic Interest selectors
  - Extend dynamic prefix registration and maintenance

L2 Communication

- Run directly over layer 2 interfaces
  - IEEE 802.15.4
    - send packets to FF:FF (broadcast)
  - Ethernet (e.g., debugging on native plaform)
    - send packets to FF:FF:FF:FF:FF:FF:FF:FF (broadcast)
- Simple hop-by-hop fragmentation if needed



Application API



```c
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;
}
```

```c
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;
}
```

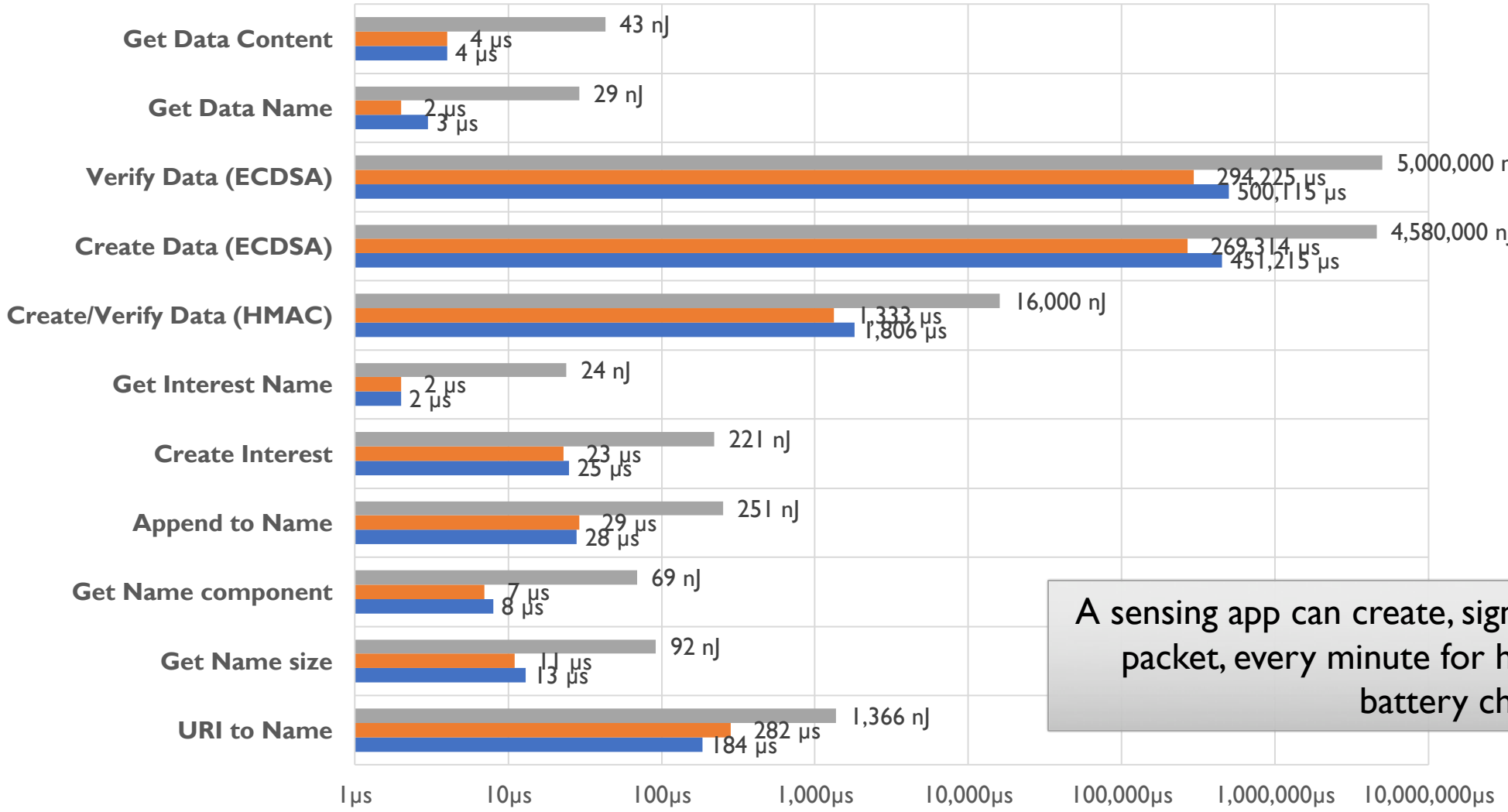
```c
static void run_client(const char* uri)
{
    handle = ndn_app_create();
    ndn_app_schedule(handle, send_interest, (void*)uri, 1000000);
    ndn_app_run(handle);
    ndn_app_destroy(handle);
}
```

Getting Started with NDN-RIOT Examples

- Downloading
  - `mkdir riot`
  - `cd riot`
  - `git clone https://github.com/named-data-iot/RIOT`
  - `git clone https://github.com/named-data-iot/ndn-riot`
  - `git clone https://github.com/named-data-iot/ndn-riot-examples`
- Compiling an example
  - cd ndn-riot-examples/<APP>
  - For host architecture (for debugging)
    - make
  - For a specific RIOT board
    - make BOARD=samr21-xpro
    - make flash BOARD=samr21-xpro # to flash firmware
    - make term BOARD=samr21-xpro # to access board via serial interface

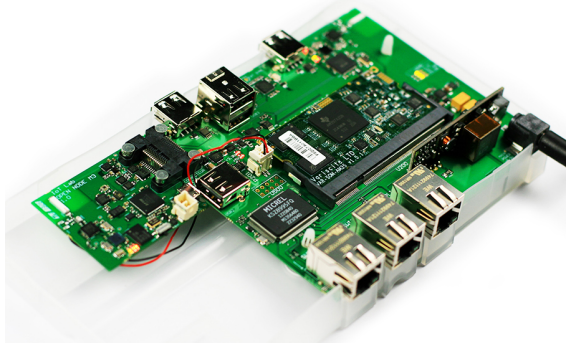
ndn-benchmark
ndn-consumer
ndn-ping
ndn-producer
ndn-rtt
ndn-template

Stack Performance Numbers



A sensing app can create, sign, and transmit one data packet, every minute for half a year on a single battery charge

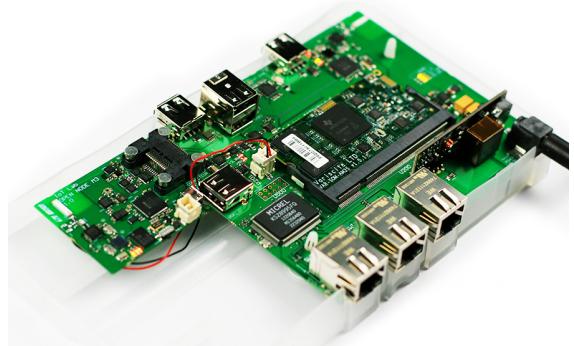
Stack Processing Delay

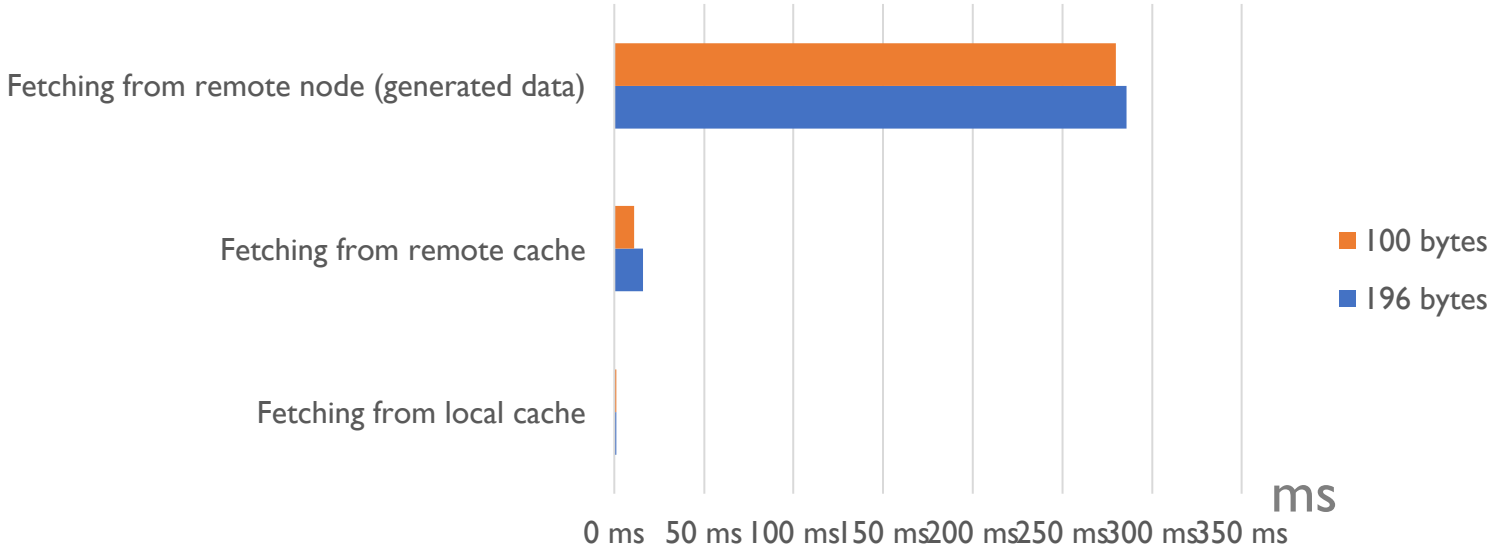


IEEE 802.15.4

MTU: 102 bytes

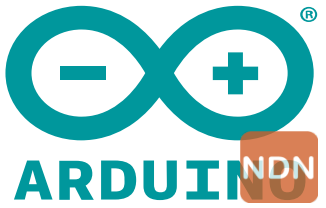
Fixed data rate: 250 Kbps



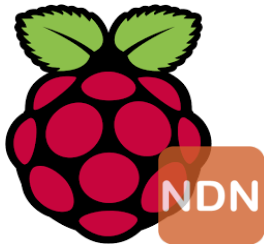


Other IoT-Related NDN Efforts

- NDN-BMS: encryption-based access control
  - Wentao Shang, Qiuhan Ding, Alessandro Marianantoni, Jeff Burke, Lixia Zhang. "Securing Building Management Systems Using Named Data Networking." In IEEE Network, Vol. 28, no. 3, May 2014.
- NDN-ACE: authorization framework for actuation apps
  - W. Shang, Y. Yu, T. Liang, B. Zhang, and L. Zhang, "NDN-ACE: Access Control for Constrained Environments over Named Data Networking," NDN Project, Tech. Rep. NDN-0036, Revision 1, December 2015.
- NDN on Arduino: minimal app for Arduino
  - https://github.com/ndncomm/ndn-btle
- https://redmine.named-data.net/projects/ndn-embedded/wiki



Options to bring NDN stack to Raspberry Pi 1/2/3



- If running actual Ubuntu repo
  - Can you PPA repository to install pre-compiled binaries
    - sudo apt-get install software-properties-common
    - sudo add-apt-repository ppa:named-data/ppa
    - sudo apt-get update
    - sudo apt-get install nfd
- If running Raspbian
  - Install from source on Pi itself (could be slow)
  - Install using cross-compilation
    - Try https://gitlab.com/named-data/docker-raspberry-pi-ndn-cross-compiler docker app
    - It is still in progress and needs feedback