Information Centric Things

Running ICN over RIOT

Marcel Enguehard
Software Engineer and PhD Candidate
RIOT Summit – July 2016
Information Centric Networking

- Each retrievable content is named
- Name-based routing
- Pull based model
- Symmetric routing
- Every node is a cache

User application

Sensor board

Interest: /roomA/temperature

Content: 20° C
Vanilla ICN Forwarding

- Forward to “faces”
- 2 modules: Forwarding Interest Base (FIB) and Pending Interest Table (PIT)
- Interest → LPM in FIB
- Content → Symmetric routing through PIT
Cisco ICN-IoT stack

• In-house implementation of ICN over RIOT

• Built for modularity and functionality:
  → based on modules (FIB, PIT, name mgmt,…)

• ~ 5k lines of code & 3 threads (main, rx/tx and beaconing)

• No cache

• W/o optimizations:
  • RAM: 23KB
  • ROM: 63KB
Dynamic forwarding

- FIB entries can point to multiple faces
- Face selection is applied through “strategies”
- e.g., Geographic forwarding

```c
typedef struct iot_fib_entry_s {
    uint8_t face_list[nb_faces];
    iot_name_t prefix;
    strategy_callback_t strategy;
} iot_fib_entry_t;
```
Geographic Forwarding in WSN

- Forward to geographic location
- Forwarding based on local properties
- Sensor-friendly
- Our flavour: GPSR
Geographic forwarding for ICThings

• **Data name**: `/g/locinf/rest/of/name`

• **FIB entry for `/g/`**:
  - Face: virtual face (all neighbours)
  - Strategy: GPSR

• **TLV for additional information**

<table>
<thead>
<tr>
<th>Bits</th>
<th>1-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
<th>32-39</th>
<th>40-47</th>
<th>48-55</th>
<th>56-63</th>
<th>64-71</th>
<th>72-79</th>
<th>80-87</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
<td>GPSR opcode</td>
<td>Length (88)</td>
<td>FLAGS</td>
<td>Perimeter entry x-coordinate</td>
<td>Perimeter entry y-coordinate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Geographic forwarding for ICThings – cont’d

- **Extract name**
- **FIB module**
  - Get faces & strategy
- **Strategy module**
  - Apply strategy

Forward on selected faces

Computes next node in GPSR

Returns virtual face + wrapper to GPSR
Beaconing for ICThings

Neighbourhood + location updates

Association: See [1]

Encryption through AES broadcast keys

Our IoT hardware

OpenMote

- ARM Cortex-M3 @ 32MHz
- AES+ECC hardware support
- 32KB RAM
- 512KB ROM
- Open source design
Evaluation setup

Cycle counter in M3

```c
#include <cc2538.h>
#include <stdint.h>

int main () {
    uint32_t nb_cycles;
    // Enables debug
    CoreDebug->DEMCR |= _VAL2FLD(CoreDebug_DEMCR_TRCENA, 1);
    // Enables cycles counter
    DWT->CTRL |= _VAL2FLD(DWT_CTRL_CYCCNTENA, 1);

    populate_tables ();

    // Reinitialises cycle counter
    DWT->CYCCNT = 0;
    perform_test ();
    nb_cycles = DWT->CYCCNT;

    ...
}
```
Forwarding vs cryptography/communication

Communication & cryptography costs estimated thanks to:
Shafagh et al. *Talos: Encrypted Query Processing for the Internet of Things, SenSys’ 15*
Going forward

Power Monitor plugged on VCC and GND pins on the OpenMote
Going *forwarder*

- **Full-scale evaluation**
  - Micro-benchmark of all energy values
  - Going fullscale: testbed/emulation?
  - Cost of control traffic

- **Limits of GPSR**
  - Optimizing neighbour count
  - Hyperbolic routing
Reflecting on RIOT

• Programming made easy
  • Avoid lots of painful things (shell, crypto, network)
  • Wiki is very helpful (thanks to the community ;-))
  • (almost) Portable code from one platform to another
  • Modules’ system
  • Native mode

• Hardware support not yet complete
  • e.g., encryption modules
GPSR – Greedy and perimeter mode

Greedy forwarding

Perimeter forwarding
Memory & CPU consumption

![Graph showing Memory & CPU consumption](image)

- CPU
- Memory
- Combined

- Number of FIB entries vs. Number of neighbours

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential