H2020 i**P** **O**ver **I**c**N**– the bet**T**er IP (POINT)

Examples

Deploy an Example COAP-over-IP-over-ICN Network

List of Authors: Hasan M A Islam, Dmitrij Lagutin

1. Introduction

2. Get Started
    2.1 Install the POINT prototype

3. Installing CoAP software components
    3.1 CoAP Client
    3.2 CoAP Server
    3.3 CoAP Proxy

4. Integrating DHT11 to Raspberry Pi

5. Examples

1. Introduction

This document describes the installation and configuration of CoAP proxy components for the POINT prototype.

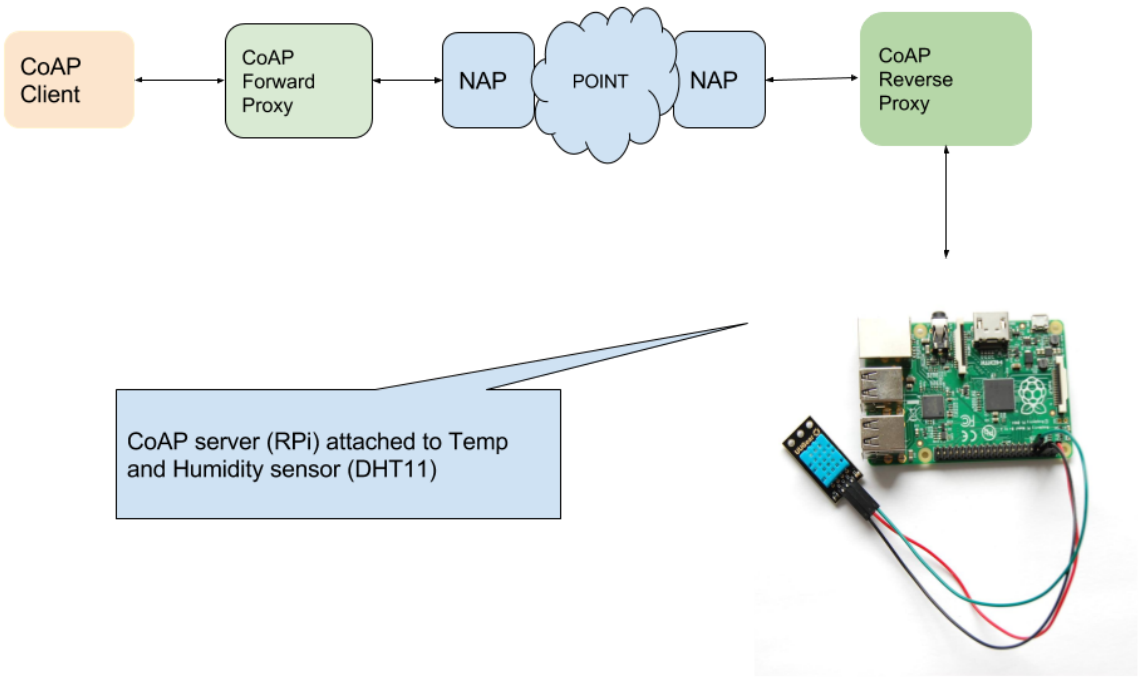

Figure 1: An example CoAP proxy setup.

2. Get Started

2.1 Install the POINT prototype

- Follow instructions as "***doc/HowTos/HowTo***" document in doc directory.

- Follow instructions as ***GetStarted*** document in the same directory.

N.B. The NAPs (shown in the middle of Figure 1) should be configured correctly in order to provide *IP-over-ICN* functionality.

3. Installing CoAP software components

3.1 CoAP Client

1. Install libcoap from Github: git clone https://github.com/obgm/libcoap
2. Install libcoap.
3. After the installation of libcoap, you can run CoAP client from /examples directory.

3.2 CoAP Server

To test CoAP over POINT setup, you can run CoAP server from /examples directory of libcoap. If you want to play with physical sensors you would need a computer such as Raspberry Pi to connect sensors to it.

The following instructions are are based on a Raspberry Pi and DHT11_sensor that can measure the temperature and humidity.

1. Install Raspbian OS in Raspberry Pi: https://www.raspberrypi.org/documentation/installation/installing-images/
2. Install microcoap from Github in Raspberry Pi: https://github.com/1248/microcoap
3. Connect the DHT11 sensor to highlighted GPIO pins of Raspberry Pi as shown in Figure 2.
4. Integrate DHT11 sensor reading module to microcoap. The details is described in Section 4.

3.3 CoAP Proxy

1. Go to apps/coapproxy directory
2. Install CoAP proxy in computers acting as a forward and reverse proxy (see Figure 1).
   ./make

At this point, you are ready to experiment with the CoAP over POINT setup.

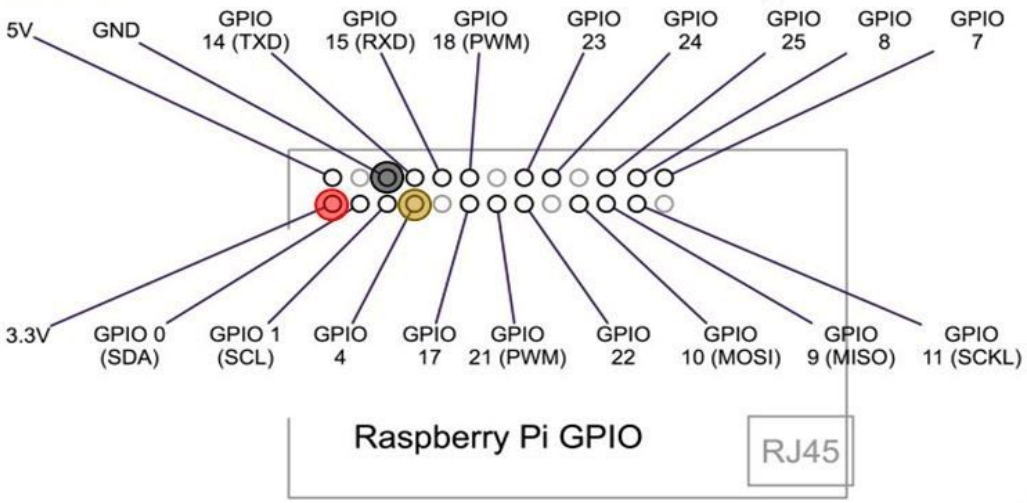

Figure 2: Raspberry Pi GPIO pins for connecting DHT11 sensor.

4. Integrating DHT11 to Raspberry Pi

Microcoap on Raspberry Pi uses the WiringPi library which is a GPIO access library written in C (http://wiringpi.com/).

1. Install WiringPi for Raspberry Pi from Github: https://github.com/WiringPi/WiringPi
2. cd wiringPi
3. ./build

This build script will compile and install it all for you – it does use the sudo command at one
point, so you may wish to inspect the script before running it.

4. Change the file **endpoints.c** in microcoap directory (the example code reads both the temperature and humidity values):

To define the resource uri of DHT11, add the following line:

```
static const coap_endpoint_path_t path_humidity = {1, {"humidity"}};
```

Add the following line:

```
{COAP_METHOD_GET, handle_get_humidity, &path_humidity, "ct=0"}
```

in const coap_endpoint_t endpoints[] array.

After adding this the statement will look like as follows:

```
const coap_endpoint_t endpoints[] =
{
  {COAP_METHOD_GET, handle_get_well_known_core, &path_well_known_core, "ct=40"},
  {COAP_METHOD_GET, handle_get_light, &path_light, "ct=0"},
  {COAP_METHOD_PUT, handle_put_light, &path_light, NULL},
  {COAP_METHOD_GET, handle_get_humidity, &path_humidity, "ct=0"},/*add this*/
  {(coap_method_t)0, NULL, NULL, NULL}
};
```

Add the following function:

```
static int handle_get_humidity(coap_rw_buffer_t *scratch,
                        const coap_packet_t *inpkt,
                        coap_packet_t *outpkt,
                        uint8_t id_hi, uint8_t id_lo)
{
  memset(rsp,'\0',1500);
  read_dht11_dat(dsp);

  return coap_make_response(scratch, outpkt,
                       (const uint8_t *)dsp,
                        strlen(dsp),
                        id_hi, id_lo, &inpkt->tok,
                        COAP_RSPCODE_CONTENT,
```

COAP_CONTENTTYPE_TEXT_PLAIN);

}

5. Copy **sensor.c sensor.h** from "apps/coapproxy/sensor" directory to the microcoap

directory on a Raspberry Pi.

6. Compile microcoap by invoking ./make
7. Run **microcoap server** with sudo.

   sudo ./coap

At this stage, the CoAP server is ready to read the sensor values.

5. Examples

We assume that the NAP has been installed correctly and is running. Each box in Figure 1 is a separate machine (or a VM). We assume that the hostnames of CoAP Reverse Proxy and Raspberry Pi are aalto.coap.point and rpi.coap.aalto. Run following commands in order to read sensor values from the CoAP client.

1. Start forward proxy: ./proxy -i aalto.coap.point -s port1
2. Start reverse proxy: ./proxy -i rpi.coap.aalto
3. Test CoAP client: ./coap-client -T cafe -P ip address of Forward Proxy Machine:port1 -v 100 -m get coap://rpi.coap.aalto/humidity