Simple Java App

Carlo Vallati
PostDoc Researcher@ University of Pisa
c.vallati@iet.unipi.it

Motivation

- Border router allows applications to access motes using their global addresses
- Next question: how to write an application that accesses data generated by sensors from applications?
- A simple UDP socket can be used from applications to interact with the sensors, see UDPClient.java

Limitation

- How to discover information available on each sensor?
- How to ask for a specific piece of information? (e.g. temperature, rather than position?)
- How to encode the information?

- **With this simple UDP solution the implementation of these functionalities are left to the programmer**
  - **Different solutions**
  - **No interoperability**

CoAP

Carlo Vallati
PostDoc Researcher@ University of Pisa
c.vallati@iet.unipi.it

CoAP

- CoAP is an application protocol similar to HTTP.
- Specifically designed for constrained environment.
- Works over UDP by default.

- It exposes functionalities provided by things as resources that can be discovered and accessed in the same way browsers access HTTP resources

Erbium

- Erbium create a CoAP server on a mote:
  - A server statically defines its resources
  - Each resource has its allowed methods
  - Each resource must be implemented statically

Copper

- Copper is a Firefox extension.
- It is a CoAP client.
- Useful to debug CoAP servers
- Can work with different CoAP version.

coap://vs0.inf.ethz.ch/

Define a CoAP Server

```
#include “contiki.h”
#include “contiki-net.h”
#include "rest-engine.h”
PROCESS_THREAD(server, ev, data){
        PROCESS_BEGIN();
        rest_init_engine();
        rest_activate_resource(&res_hello, “hello”);
        while(1) {
                PROCESS_WAIT_EVENT();
        }
        PROCESS_END();
}
```

Define a resource

RESOURCE(name_resource, attributes, get_handler, post_handler, put_handler, delete_handler);

void

get_handler(void* request, void* response, uint8_t *buffer, uint16_t preferred_size, int32_t *offset){

…

REST.set_header_content_type(response, REST.type.TEXT_PLAIN);
REST.set_header_etag(response, (uint8_t *) &length, 1);

REST.set_response_payload(response, buffer, length);

}

Makefile

```
UIP_CONF_IPV6=1
CFLAGS += -DUIP_CONF_IPV6=1
CFLAGS += -DUIP_CONF_IPV6_RPL=1
CONTIKI=../..

CFLAGS += -DPROJECT_CONF_H=\"project-conf.h\"
CFLAGS += -DUIP_CONF_TCP=0

APPS += er-coap
APPS += rest-engine
```

Project-conf

```
#undef IEEE802154_CONF_PANID
#undef NETSTACK_CONF_RDC
#define NETSTACK_CONF_RDC     nullrdc_driver
#undef NETSTACK_CONF_MAC
#define NETSTACK_CONF_MAC     nullmac_driver
#undef REST_MAX_CHUNK_SIZE
#define REST_MAX_CHUNK_SIZE    64
#undef COAP_MAX_OPEN_TRANSACTIONS
#define COAP_MAX_OPEN_TRANSACTIONS   4

/* Save some memory for the sky platform. */
#undef NBR_TABLE_CONF_MAX_NEIGHBORS
#define NBR_TABLE_CONF_MAX_NEIGHBORS     10
#undef UIP_CONF_MAX_ROUTES
#define UIP_CONF_MAX_ROUTES   10
#undef UIP_CONF_BUFFER_SIZE
#define UIP_CONF_BUFFER_SIZE    280
```

Exercise 1

- Deploy a CoAP server with a only one resource.
- The resource must allow the GET method.
- Use Copper to interact with the CoAP server. Try CON and NON messages.

- NOTE: in order to interact between Copper (running on the host) and the CoAP server a border router is needed.

- er-server-only-get.c

Change the value of a resource

- Retrieve method:
  - uint8_t method = REST.get_method_type(request);
- Check method:
  - if (method & METHOD_POST)
- Set response:
  - REST.set_response_status(response, REST.status.CREATED);
  - REST.set_response_status(response, REST.status.BAD_REQUEST);

Parameters

- Get a query parameter (**URL?value=10**):
  - REST.get_query_variable(request, "color", &color)
- Get a post parameter (**value=10 in the post or put payload**):
  - REST.get_post_variable(request, "mode", &mode)
- Analyze parameter:
  - strncmp(mode, "on", len)

Exercise 2

- Add a resource that accepts both GET and PUT requests to retrieve and set an integer value, respectively.
- If the client sends a PUT request the server must update the value. Post parameter “value=20”
- The GET must return the stored value.

- er-server-put.c

Exercise 3

- Write a CoAP server with a resource which change the status of the leds depending on query and post parameters.
- Query parameter:
  - color=r|g|b
- Post parameter:
  - mode=on|off

- er-leds.c

Californium

Carlo Vallati
PostDoc Researcher@ University of Pisa
c.vallati@iet.unipi.it

Californium

- Californium is a Java CoAP library
- Easy and simple interface to deploy CoAP enabled applications
- Download with:
  - git clone https://github.com/eclipse/californium.git
- Compile using Maven:
  - mvn install

  or
  - Eclipse:
    - Download Maven plugins
    - import Maven project

GET Example

```
URI uri = null;

try{
		uri = new URI("coap://[aaaa::c30c:0:0:9e]:5683/hello");
} catch (Exception e) {
		System.err.println("Caught Exception: " + e.getMessage());
}

CoapClient client = new CoapClient(uri);

CoapResponse response = client.get();

if (response!=null) {

		System.out.println(response.getCode());
		System.out.println(response.getOptions());
		System.out.println(response.getResponseText());

		System.out.println("\nADVANCED\n");
		// access advanced API with access to more details through .advanced()
		System.out.println(Utils.prettyPrint(response));

} else {
		System.out.println("No response received.");
}
```