Observing

Giacomo Tanganelli
PhD student @ University of Pisa
g.tanganelli@iet.unipi.it

Contiki - Observing

- The Contiki implementation is resource centric.
- To enable observing on a resource, the resource itself must be observable.
- Contiki defines also periodic resources. The handler is called periodically.
- Two type of observable resources:
  - EVENT_RESOURCE
  - PERIODIC_RESOURCE

Event resources

- Define the resource:
  - EVENT_RESOURCE(resource1, METHOD_GET, “resource1", "title=\“Resource\";obs");
- Define the standard handler:
  - resource1_handler(…)
  - Responses to GET requests
- Define the event handler:
  - void resource1_event_handler(…)
  - Response to events

Event handler

```
void resource1_event_handler(resource_t *r){

...
/* Build notification. */
  coap_packet_t notification[1];
  coap_init_message(notification, COAP_TYPE_CON,
REST.status.OK, 0 );
  coap_set_payload(notification, content, snprintf(content,
sizeof(content), "EVENT %u", event_counter));


REST.notify_subscribers(r, event_counter, notification);
}
```

Activate event resource

```
rest_activate_event_resource(&resource_resource1);

 while(1) {
   PROCESS_WAIT_EVENT();
   if (ev == sensors_event && data == &button_sensor) {
         resource1_event_handler(&resource_resource1);
   }
}
```

Periodic resources

- Define the resource:
  - PERIODIC_RESOURCE(resource2, METHOD_GET, “resource2", "title=\"Periodic demo\";obs", 5*CLOCK_SECOND);
- Define the standard handler:
  - resource2_handler(…)
  - Responses to GET requests
- Define the periodic handler:
  - void resource2_periodic_handler(…)
  - Response periodically

Periodic handler

```
void resource2_periodic_handler(resource_t *r){
...
/* Build notification. */
 coap_packet_t notification[1];
 coap_init_message(notification, COAP_TYPE_NON,
REST.status.OK, 0 );
  coap_set_payload(notification, content, snprintf(content,
sizeof(content), "TICK %u", obs_counter));

REST.notify_subscribers(r, obs_counter, notification);
}
```

Activate periodic resource

```
rest_activate_periodic_resource(&resource_resource2);
```

Important:

If the notification will be sent only as NON messages a server can not understand if a client is still alive.

To overcome sends a CON message every X NON messages.

Exercise 1

- Write a CoAP server with two resources:
  - The former must be an event resource which will send a CON notification to observers when the server button is pressed.
  - The latter is a periodic resource which sends a NON notification every 10 seconds.

Content-type

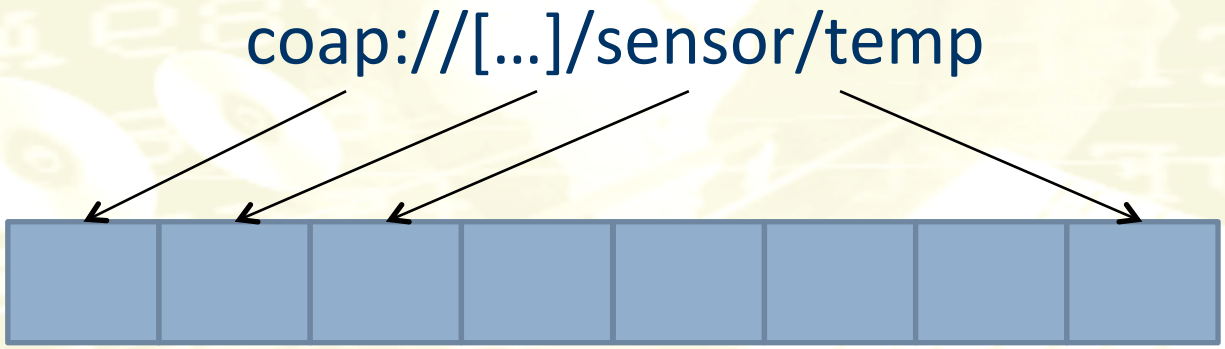
- In order to exchange information correctly CoAP defines Content-type.
- A client can set an Accept option in a request.
- The server try to response with an accepted content-type.
- To retrieve accept options:
  - const uint16_t *accept = NULL;
  - int num=REST.get_header_accept(request, &accept);

Exercise 2

- Modify the previous example in order to send notification in the xml format.
- Remember to set the content-type of the response accordingly:
  - REST.set_header_content_type(response, REST.type.APPLICATION_XML);

ETag

- The Etag is used to validate responses:
  - A resource representation is characterized by an Etag.



ETag

- When a client obtains a response it also obtains an Etag.
- When the client wants an update send a request with also the Etag obtained previously.
- If the resource representation is still valid the server reply with 2.03 valid without include in the payload the resource representation (the client already has the most updated one)

Contiki ETag

- To retrive the Etag contained in a request:
  - const uint8_t *bytes = NULL;
  - len = coap_get_header_etag(request, &bytes);
- To set an Etag:
  - REST.set_header_etag(response, etag, etag_len);
- To set 2.03 code:
  - REST.set_response_status(response, REST.status.NOT_MODIFIED);

Exercise 3

- Defines a resource which reply to GETs with also an ETag. If the request has the correct Etag the server must reply wit 2.03.
- When the client sends a POST message the resource must be updated with the content of the request and the Etag must change.