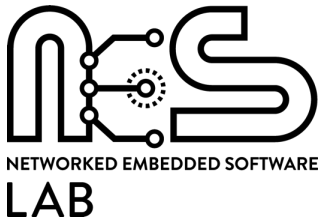


POLITECNICO
MILANO 1863

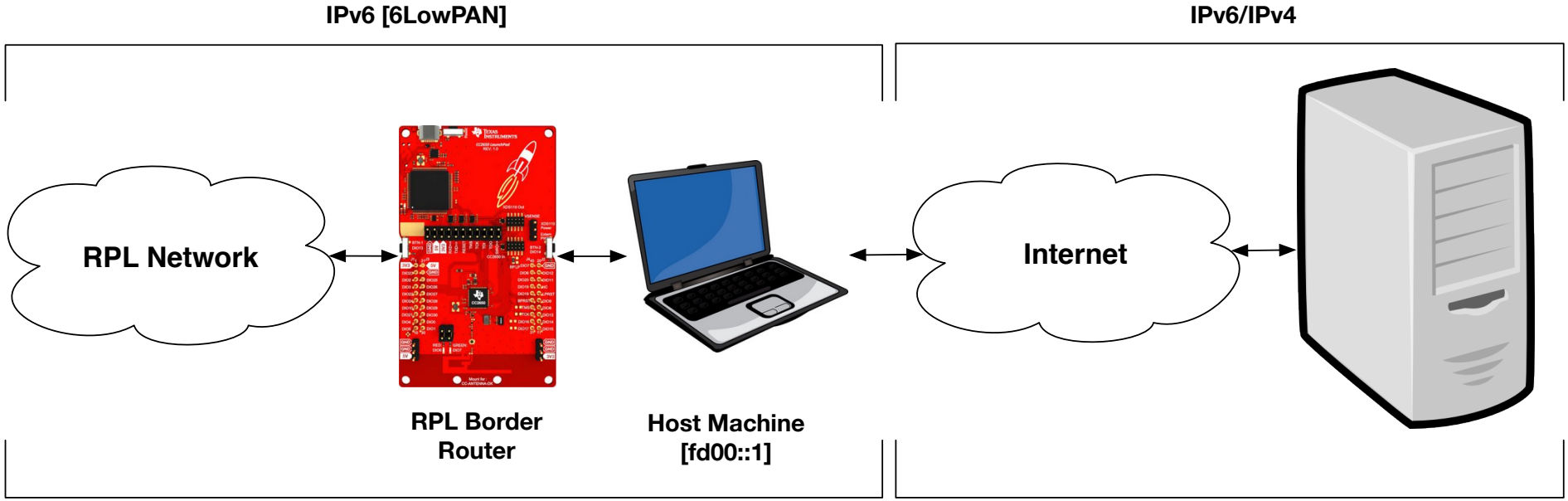
RI.
SE

Contiki-NG
MQTT

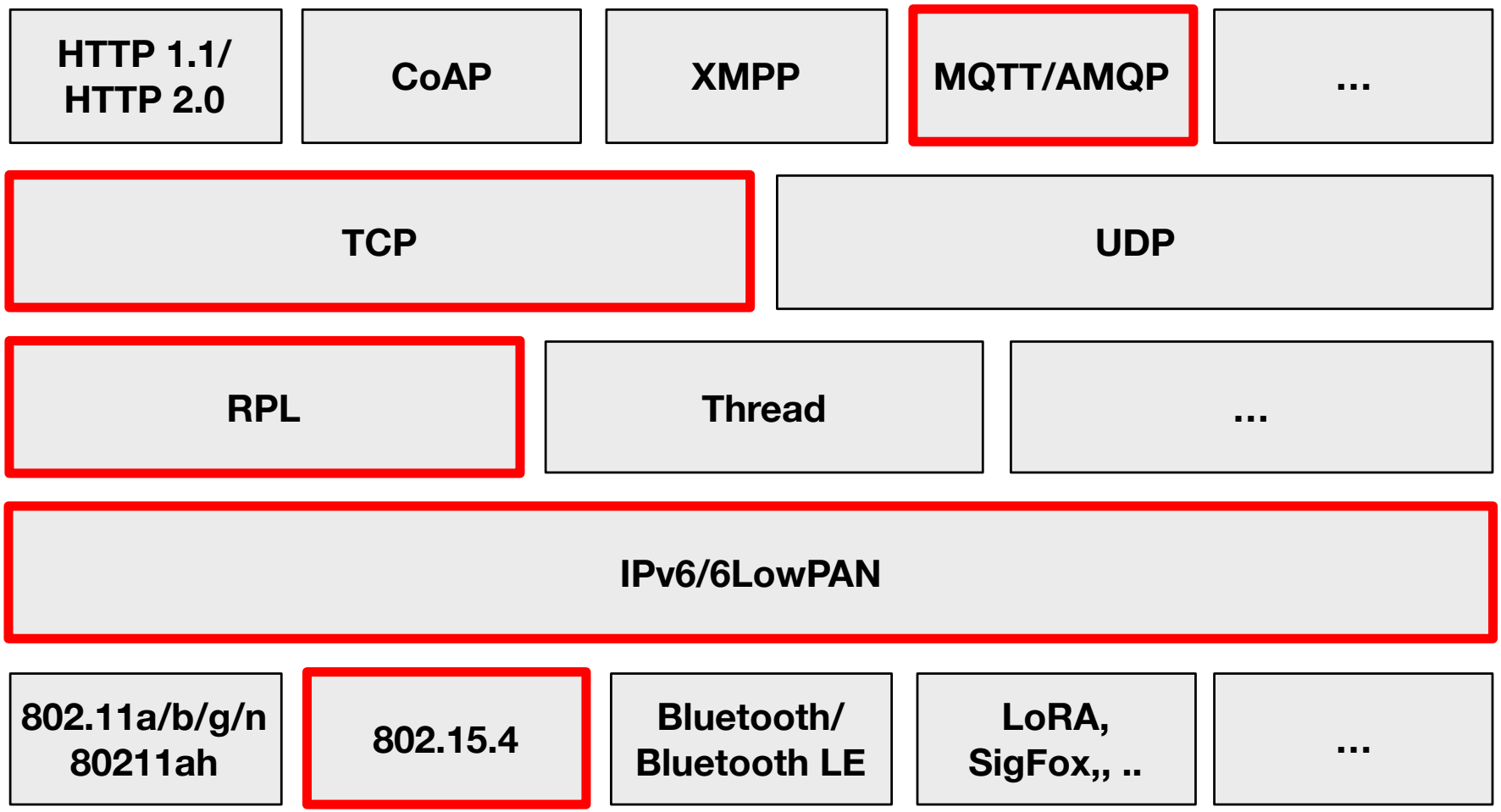
Luca Mottola
luca.mottola@polimi.it

RPL Border Router

- We need a “man-in-the-middle” bridging the IoT network with the external Internet
- In Contiki-NG: **`examples/rpl-border-router`**
  - Configured to also become the root of the RPL tree



Complete Stack Configuration



MQTT in Contiki-NG

- Implementation compliant with v.3.1
- Does not support QoS2
- Contiki-NG devices can operate as publisher, subscriber, or both

MQTT: API (part 1)

Notify of important MQTT events, such as connection open, acknowledgement after publish if using QoS1, …

```
typedef void (*mqtt_event_callback_t)(struct mqtt_connection *m,
                                      mqtt_event_t event,
                                      void *data);

mqtt_status_t mqtt_register(struct mqtt_connection *conn,
                            struct process *app_process,
                            char *client_id,
                            mqtt_event_callback_t event_callback,
                            uint16_t max_segment_size);
```

Setup the connection to the MQTT broker

MQTT: API (part 2)

```
mqtt_status_t mqtt_connect(struct mqtt_connection *conn,
                           char *host,
                           uint16_t port,
                           uint16_t keep_alive);


mqtt_status_t mqtt_subscribe(struct mqtt_connection *conn,
                             uint16_t *mid,
                             char *topic,
                             mqtt_qos_level_t qos_level);


mqtt_status_t mqtt_publish(struct mqtt_connection *conn,
                           uint16_t *mid,
                           char *topic,
                           uint8_t *payload,
                           uint32_t payload_size,
                           mqtt_qos_level_t qos_level,
                           mqtt_retain_t retain);
```

Establish the connection to the broker

Subscribe to a topic

Publish a message

MQTT: Example (1)

- Lives in **`examples/mqtt-demo`**
- Connects to an MQTT broker running at **`fd00::1`**
- Periodically publishes messages to **`iot/native/launchpad/json`**
- Subscribes to **`iot/native/launchpad/json`**
- Published messages include meta-data and a fake temperature reading

MQTT: Example (2)

- If you are bridging to `mqtt.neslab.it`...