TP FIT/IoT-Lab using sensors

TP#1 using FIT/IoT-Lab

Lecturer: Keun-Woo Lim

Lecture slides for COMASIC

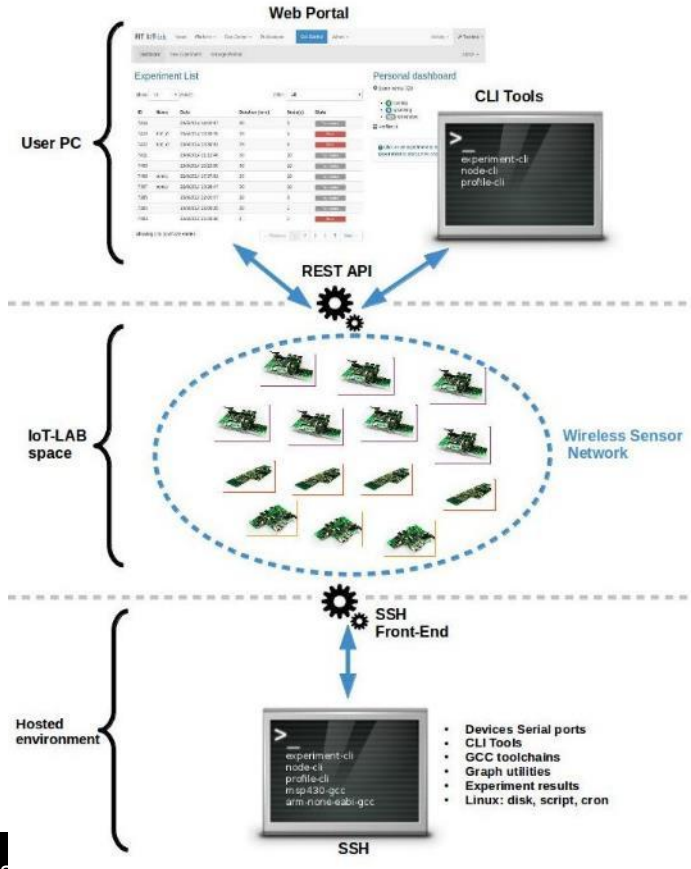
20-10-2020

What to do today

- **1. Recheck the overall architecture of FIT/IoT-lab**
- **2. Check your accounts to see if you can connect to the Web interface**
- **3. Register your local computer to connect to SSH**
- **4. Do the tutorials**
- **5. Do the challenges**
  - Become familiar with the codes

Overview of the architecture

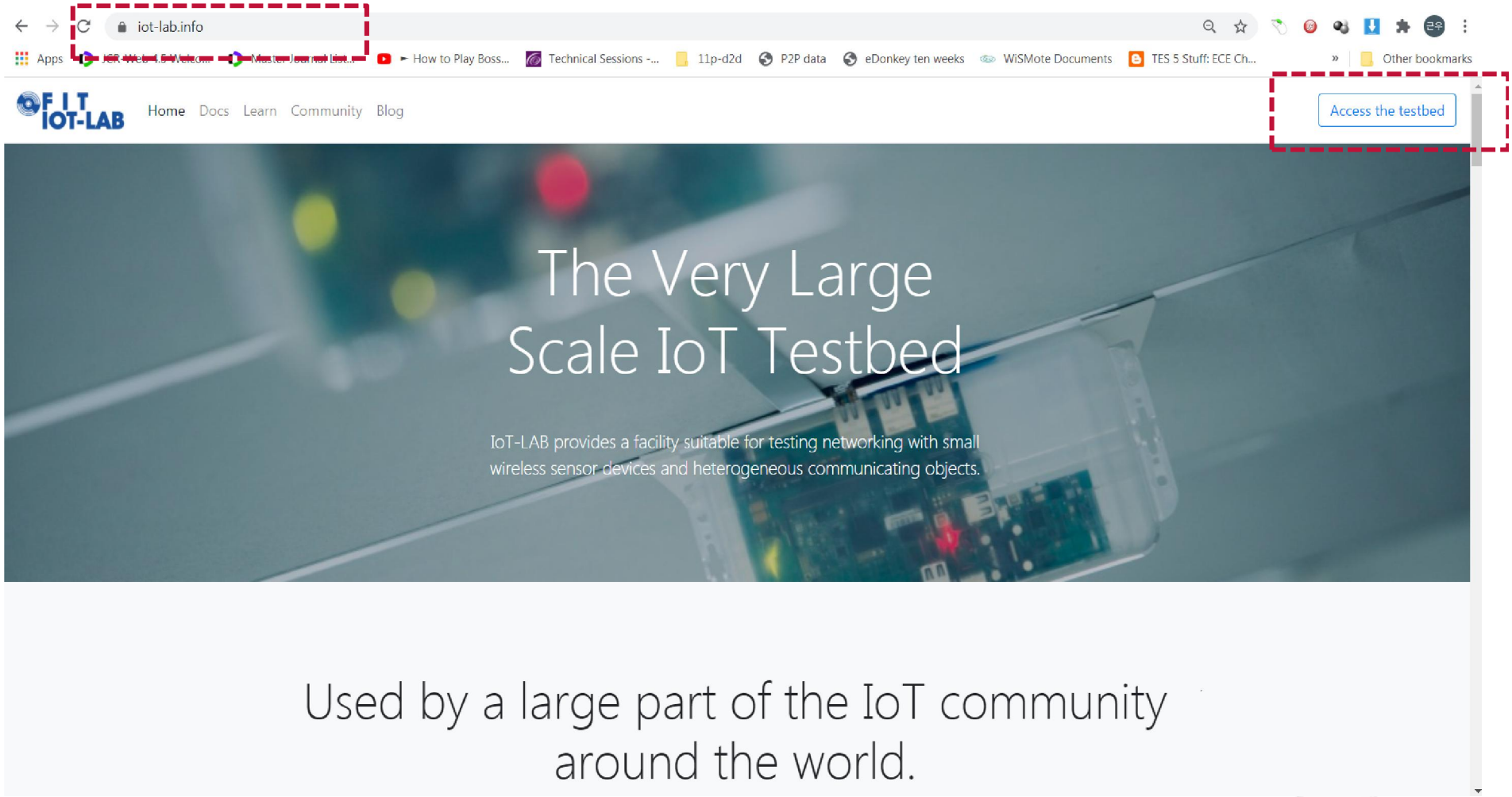
- **https://www.iot-lab.info/docs/getting-started/introduction/**



Sensor hardware

- **M3**
  - https://www.iot-lab.info/docs/boards/iot-lab-m3/
- **A8**
  - https://www.iot-lab.info/docs/boards/iot-lab-a8-m3/

Login


iot-lab.info
Apps
How to Play Boss...
Technical Sessions -...
11p-d2d
P2P data
eDonkey ten weeks
WiSMote Documents
TES 5 Stuff: ECE Ch...
Other bookmarks
F I T IOT-LAB
Home
Docs
Learn
Community
Blog
Access the testbed
The Very Large Scale IoT Testbed
IoT-LAB provides a facility suitable for testing networking with small wireless sensor devices and heterogeneous communicating objects.
Used by a large part of the IoT community around the world.

SSH access

- **Objective**
  - Connect to the front-end of FIT/IoT-Lab
  - Get access to the sensors

- **Let's do it together!**
  - https://www.iot-lab.info/docs/getting-started/ssh-access/

- **For people using Windows:**
  - https://www.iot-lab.info/legacy/tutorials/ssh-access/index.html

Be careful!

- **Know your environment!**
  - Are you working on
    - Grenoble?
    - Lille?
    - Others?
  - What is the difference?
    - Did you check the IP address?
    - Are you clearly seeing your node reserved?

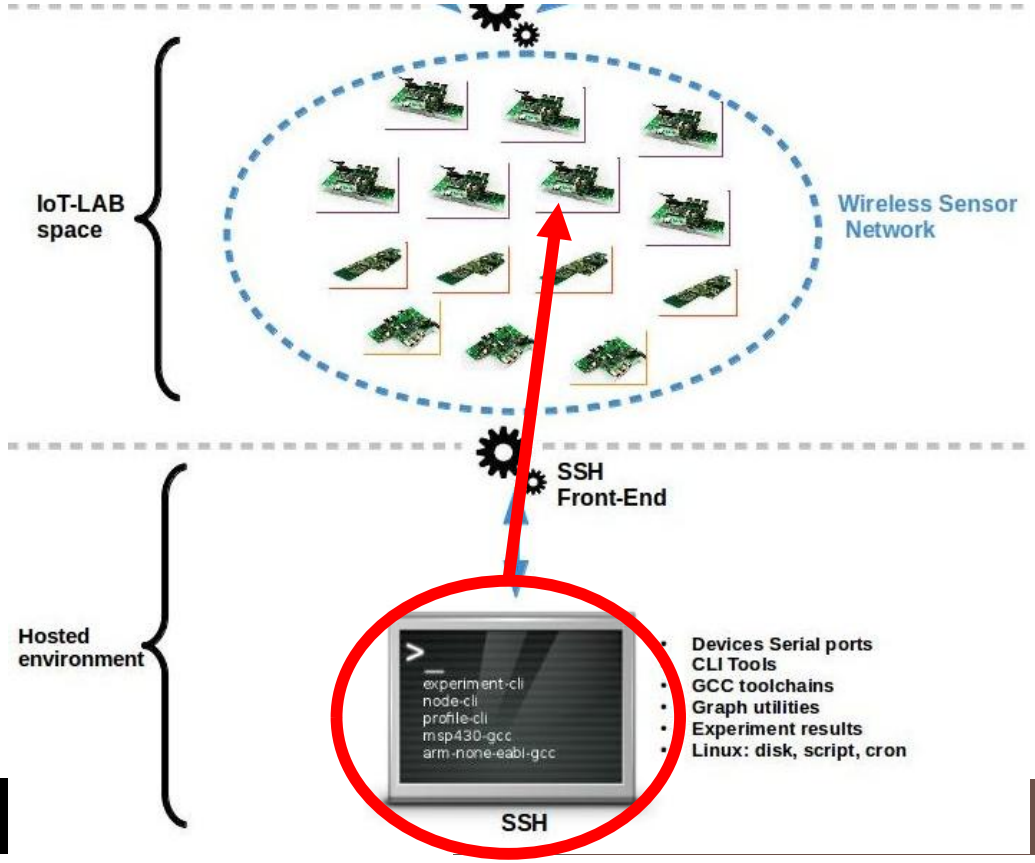
Tutorial – Sensor read

- **Objective**
  - Examine and compile Contiki
  - Create binaries for M3 nodes
  - Port binary to the sensor node
  - Read the sensings from the node

- **Let's try it together!**
  - https://www.iot-lab.info/legacy/tutorials/contiki-compilation/index.html
  - Let's understand the code!

Questions here

- **What is exactly happening?**
  - Connection using Front-end
  - No wireless communication yet



Simple Challenge

- **Try to apply these things, recompile, and experiment:**
  - Change the time period to 5 seconds
  - Print only the light information (lux)

LEDS

- **You can toggle LEDs in the sensor**
  - LEDS_RED / LEDS_GREEN / LEDS_YELLOW
- **Several functions**
  - leds_on(LEDS_RED) / leds_off(LEDS_RED)
  - leds_toggle(LEDS_GREEN)
  - leds_on(LEDS_ALL)
  - leds_get() // print in %X format

LEDS challenge

- **Read LED information and toggle them in your code**

```
LEDS state = 7
LEDS state = 0
LEDS state = 5
LEDS state = 6
LEDS state = 3
LEDS state = 1
^C
```

```
leds_off(LEDS_ALL);
printf("LEDS state = %X\n", leds_get());
leds_toggle(LEDS_ALL);
printf("LEDS state = %X\n", leds_get());
leds_off(LEDS_ALL);
leds_on(LEDS_RED);
printf("LEDS state = %X\n", leds_get());
leds_off(LEDS_RED);
leds_on(LEDS_GREEN);
printf("LEDS state = %X\n", leds_get());
leds_off(LEDS_GREEN);
leds_on(LEDS_YELLOW);
printf("LEDS state = %X\n", leds_get());
leds_on(LEDS_RED);
printf("LEDS state = %X\n", leds_get());
```

LEDS control – First taste of automation

- **Create an algorithm where:**
  - Your LED reacts to the value of the light sensor
  - Maintain information on light value
  - If there is a change in the integer value of light sensor, turn all LED on
  - If there is no change, turn of LED

```
light: 634.918213 lux
LEDS state = 7
light: 634.979248 lux
LEDS state = 7
light: 635.284424 lux
LEDS state = 0
light: 635.620117 lux
LEDS state = 7
light: 635.345459 lux
LEDS state = 7
light: 635.421753 lux
LEDS state = 7
^C
```

TP FIT/IoT-Lab
Communication

What to do

- **Enable communication between devices**
  - HTTP
- **Based on this, do the challenges**

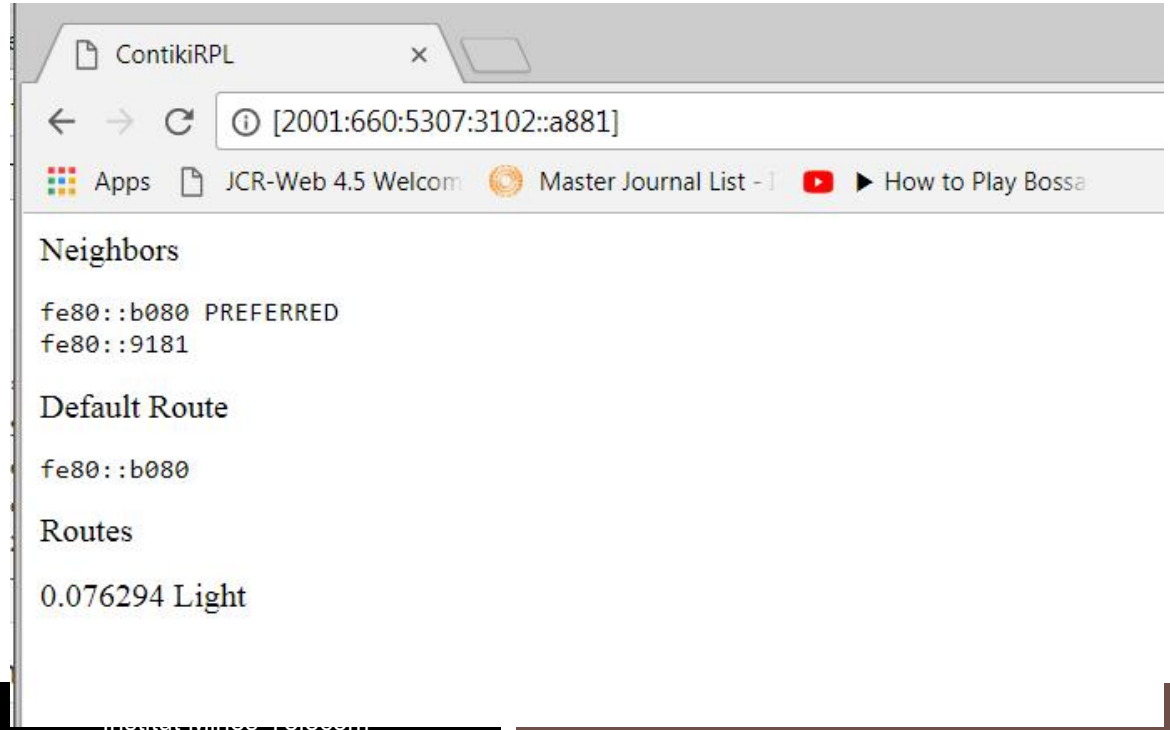
Tutorial – Public IPv6

- **Objective**
  - Create a public HTTP network where you can connect from the Internet
  - Check the function of RPL

- **Let's try it together!**
  - https://www.iot-lab.info/legacy/tutorials/contiki-public-ipv6-m3/index.html/

HTTP tutorial

- **To know you have succeeded,**
  - Open any web browser and put in:
  - http://[2001:660:5307:XXXX::YYYY]
  - XXXX = your subnet, YYYY = one of the HTTP servers



Questions here

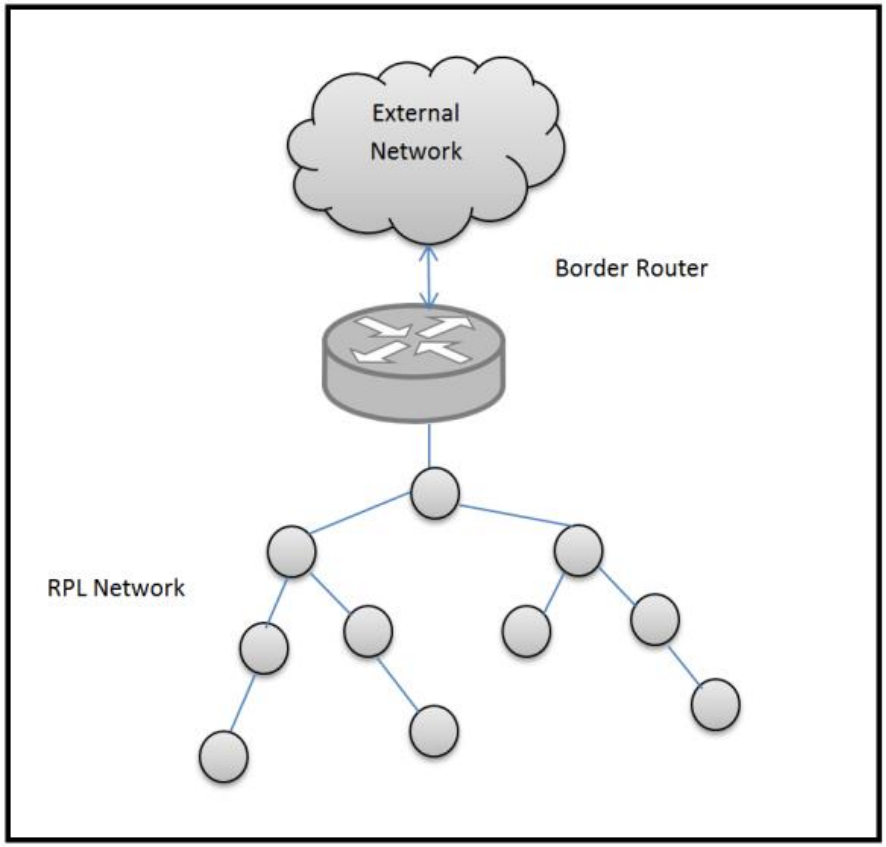
- **What is a Border Router?**
- **What is a HTTP Server?**
- **Why do we need to find an available IPv6 Prefix?**
- **What is a turnslip?**

- **These are all needed for you to connect to the sensor via Internet!!**
  - Makes it seen from outside

- **If there is a HTTP server…**
  - You can see it from a browser!!

Border router

- **What is a Border Router?**
  - Access point to internal and external network



HTTP server

- **An entity that accepts HTTP based requests from the Internet**
  - Based on TCP

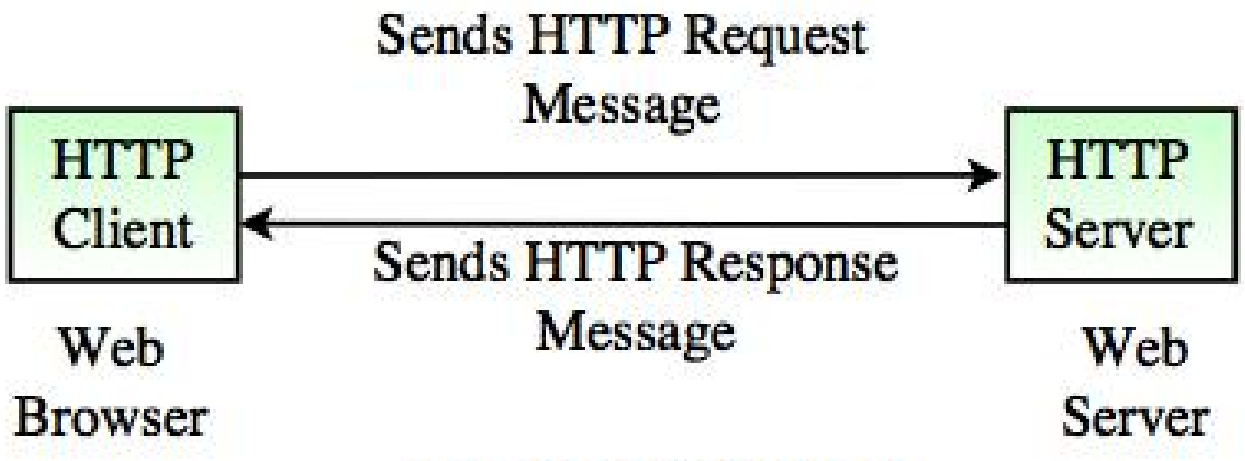
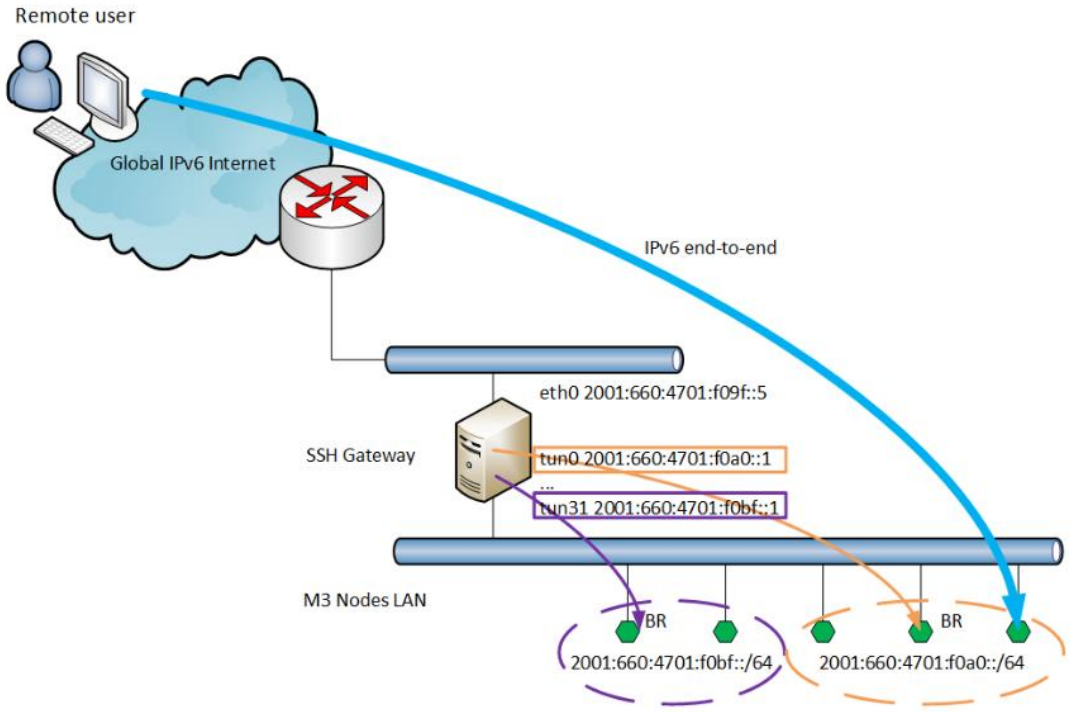


Fig. HTTP Protocol

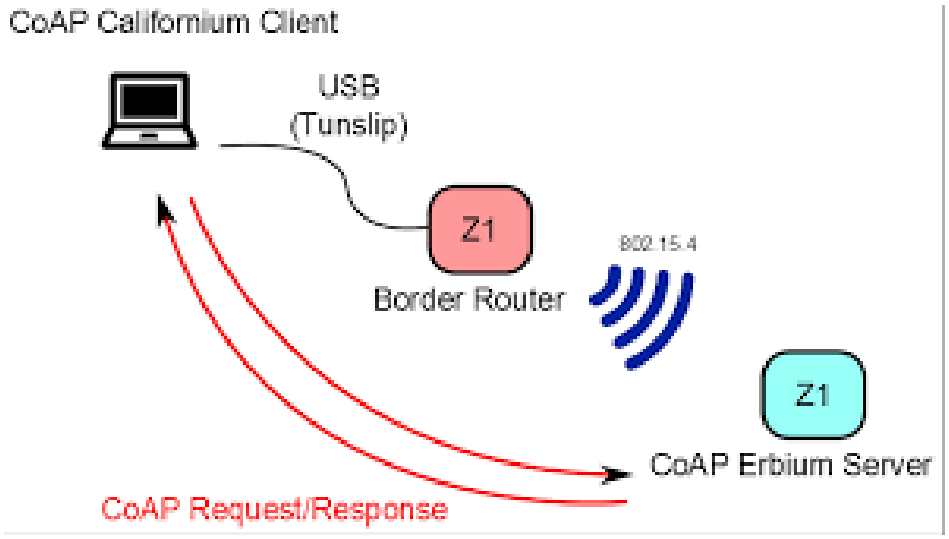
IPv6 Prefixes

- **Needed for groups of entities close to each other, use of prefixes can cluster them and make them easier to find**



Turnslip

- **Tool used to bridge IP traffic between a host and another network element, typically a border router, over a serial line.**



Challenge for today

- **Integrate HTTP and sensor-collecting!**
  - /iot-lab/parts/Contiki/examples/ipv6/http-server
  - /iot-lab/parts/Contiki/examples/iotlab/03-sensors-collecting

- **GOAL**
  - Create sensor readings from the http-server
  - Use your web browser to get sensor readings from the Internet

  - For this, let's analyze the http-server code together!

Example of HTTP-server code

```
114     ADD("</pre>\nDefault Route<pre>\n");
115     SEND_STRING(&s->sout, buf);
116     blen = 0;
117     ipaddr_add(uip_ds6_defrt_choose());
118     ADD("\n");
119     ADD("</pre>Routes<pre>");
120     SEND_STRING(&s->sout, buf);
121     blen = 0;
122     for(r = uip_ds6_route_head(); r != NULL; r = uip_ds6_route_next(r)) {
123       ipaddr_add(&r->ipaddr);
124       ADD("/%u (via ", r->length);
125       ipaddr_add(uip_ds6_route_nexthop(r));
126       if(1 || (r->state.lifetime < 600)) {
127         ADD(") %lus\n", (unsigned long)r->state.lifetime);
128       } else {
129         ADD(")\n");
130       }
131       SEND_STRING(&s->sout, buf);
132       blen = 0;
```

Example of sensor code

```
/* Light sensor */
static void config_light()
{
  light_sensor.configure(LIGHT_SENSOR_SOURCE, ISL29020_LIGHT__AMBIENT);
  light_sensor.configure(LIGHT_SENSOR_RESOLUTION, ISL29020_RESOLUTION__16bit);
  light_sensor.configure(LIGHT_SENSOR_RANGE, ISL29020_RANGE__1000lux);
  SENSORS_ACTIVATE(light_sensor);
}
static void process_light()
{
  int light_val = light_sensor.value(0);
  float light = ((float)light_val) / LIGHT_SENSOR_VALUE_SCALE;
  printf("light: %f lux\n", light);
}
```