

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/344725144>

Tutorial: Contiki-Cooja & FIT IoT-LAB

Presentation · October 2020

DOI: 10.13140/RG.2.2.30701.41442

CITATIONS

0

READS

1,088

1 author:



[Alakesh Kalita](#)

Singapore University of Technology and Design

28 PUBLICATIONS 118 CITATIONS

SEE PROFILE

Internet of Things

Tutorial on Contiki OS, Cooja and FIT IoT-LAB

Alakesh Kalita

PhD Scholar

Indian Institute of Technology Guwahati

E-mail: alakesh.kalita025@gmail.com

What is Contiki OS ?

- Contiki is an **open source** **operating system** for **Internet of Things**
 - runs on **tiny low-power** **microcontrollers**
- It **allows to develop applications** that make efficient use of different hardware for IoT
 - while providing **standardized** low-power wireless communication for a range of hardware platforms
 - Mainly, focus on **low-power** **wireless** IoT devices
- The Contiki system includes a **sensor simulator** called **Cooja**,
 - Cooja **simulates** of Contiki nodes

How to install Contiki-NG

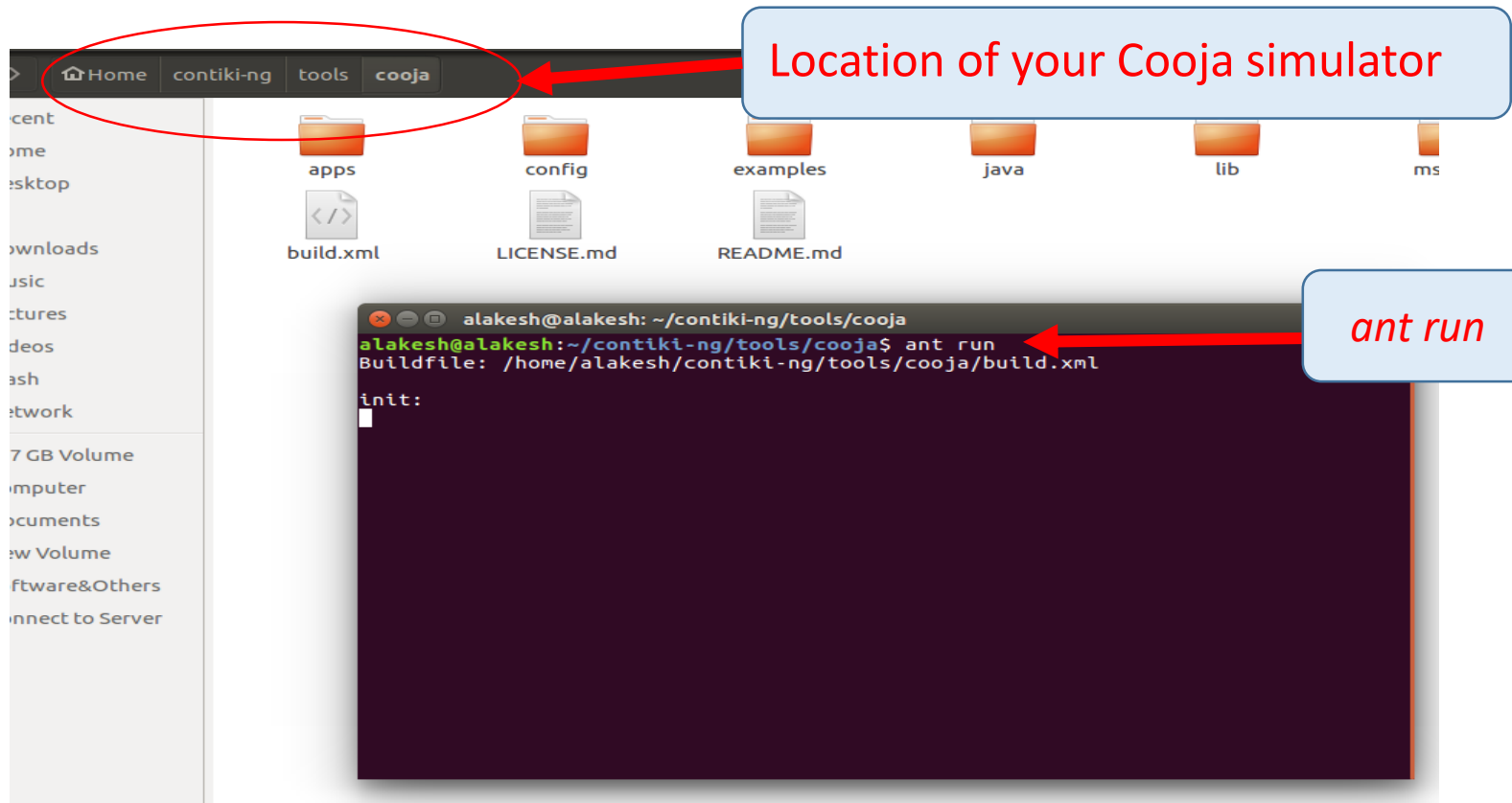
- The latest version of Contiki is known as **Contiki-NG** (Contiki Next Generation)
- Complete installation procedure of Contiki-NG on Linux can be found at the below link,
 - <https://github.com/contiki-ng/contiki-ng/wiki/Toolchain-installation-on-Linux>
- One can install Contiki-NG using **Virtual Box** on Windows OS.

What is Cooja?

- Cooja is a Contiki **network simulator**
 - To perform **IoT network simulations**
 - An extensible **Java-based simulator** capable of emulating various IoT motes such as Tmote sky, Z1 etc.,
 - The code to be executed by the node is the exact **same firmware** that can be **uploaded to physical nodes**
 - Allows **large and small networks** of motes can be simulated at **hardware level**
- Cooja is a highly useful tool for Contiki development
 - It allows developers to **test their code and systems** before running it on the target hardware

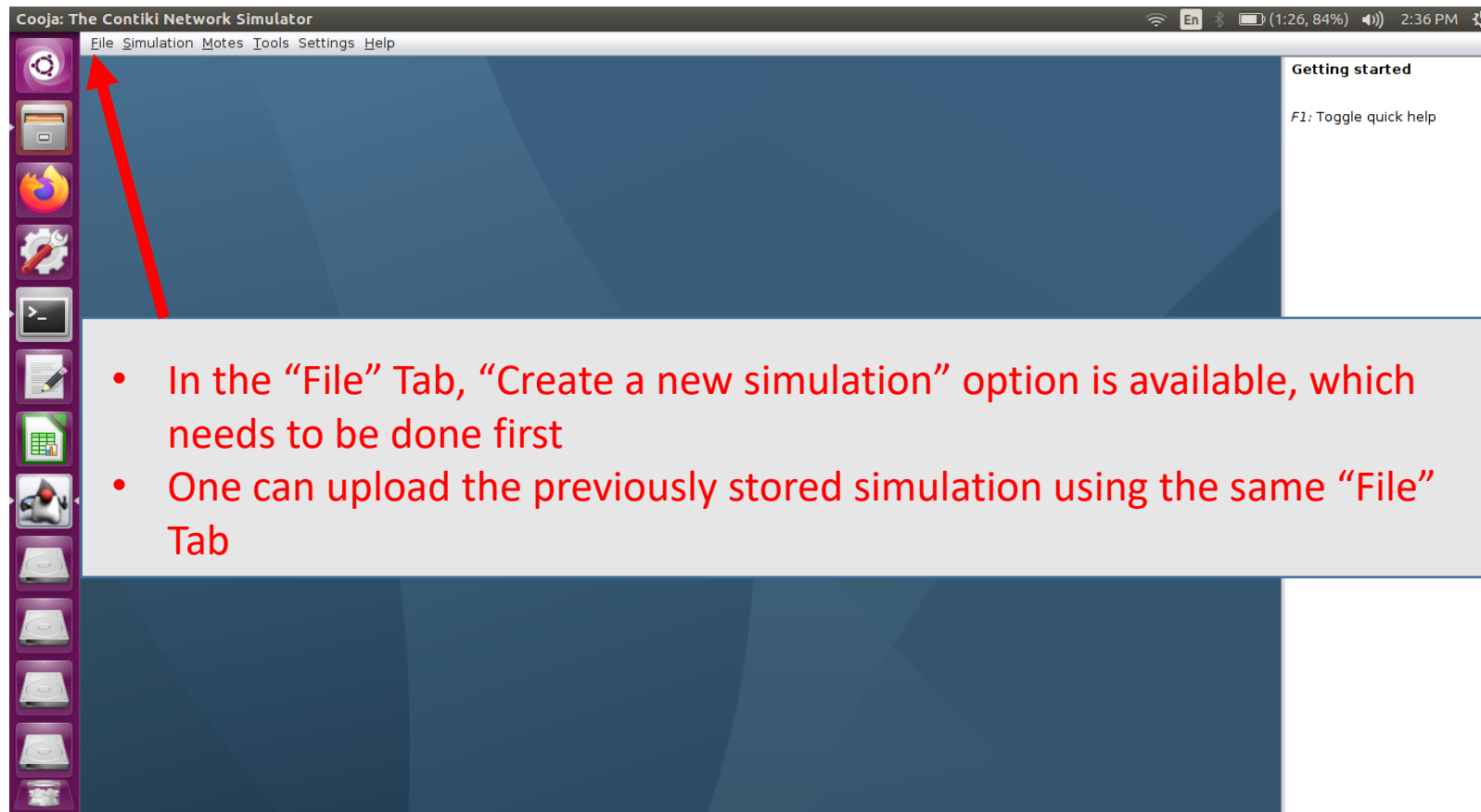
How to start Cooja?

- After Contiki-NG installation, **start** Cooja Simulator using the command “*ant run*” inside the **Cooja directory** of Contiki-NG.

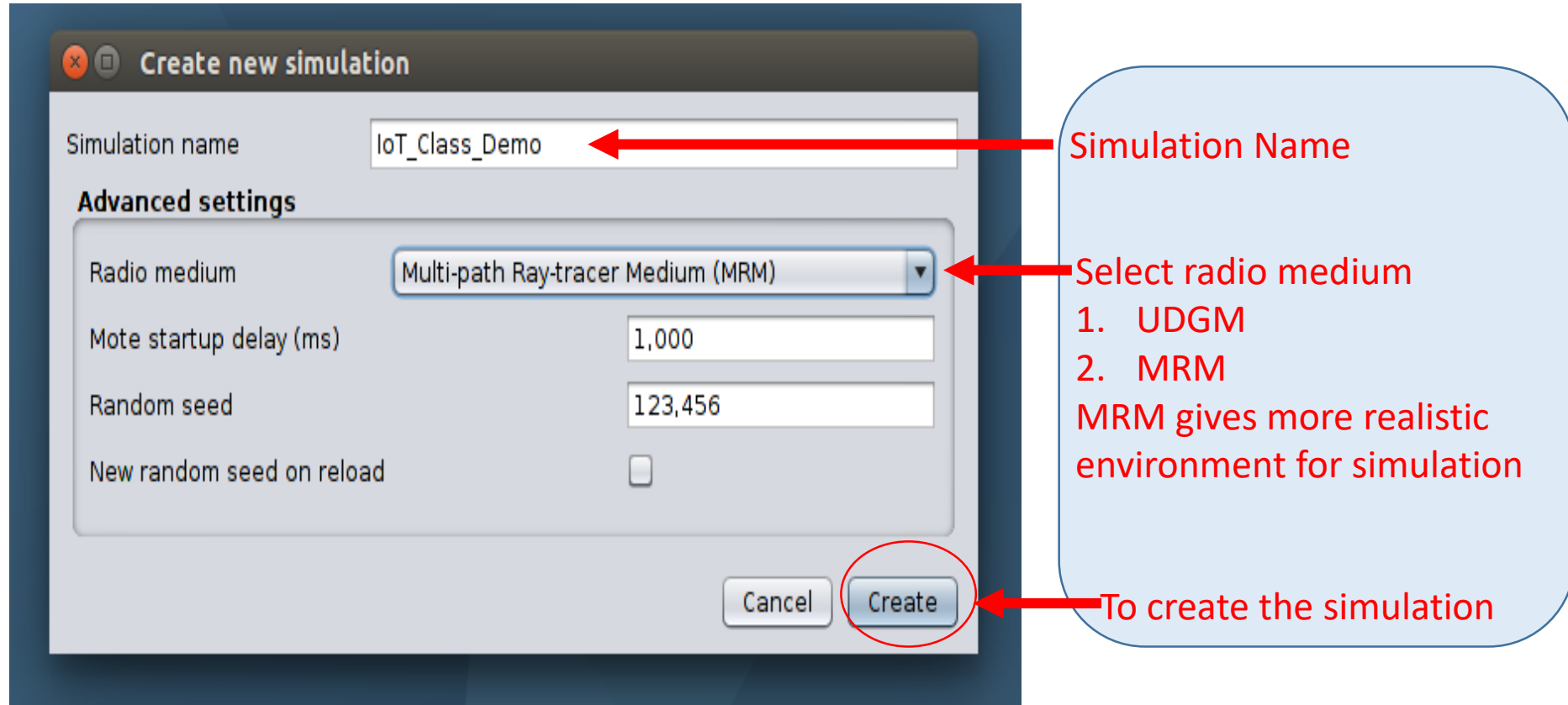


Contd..

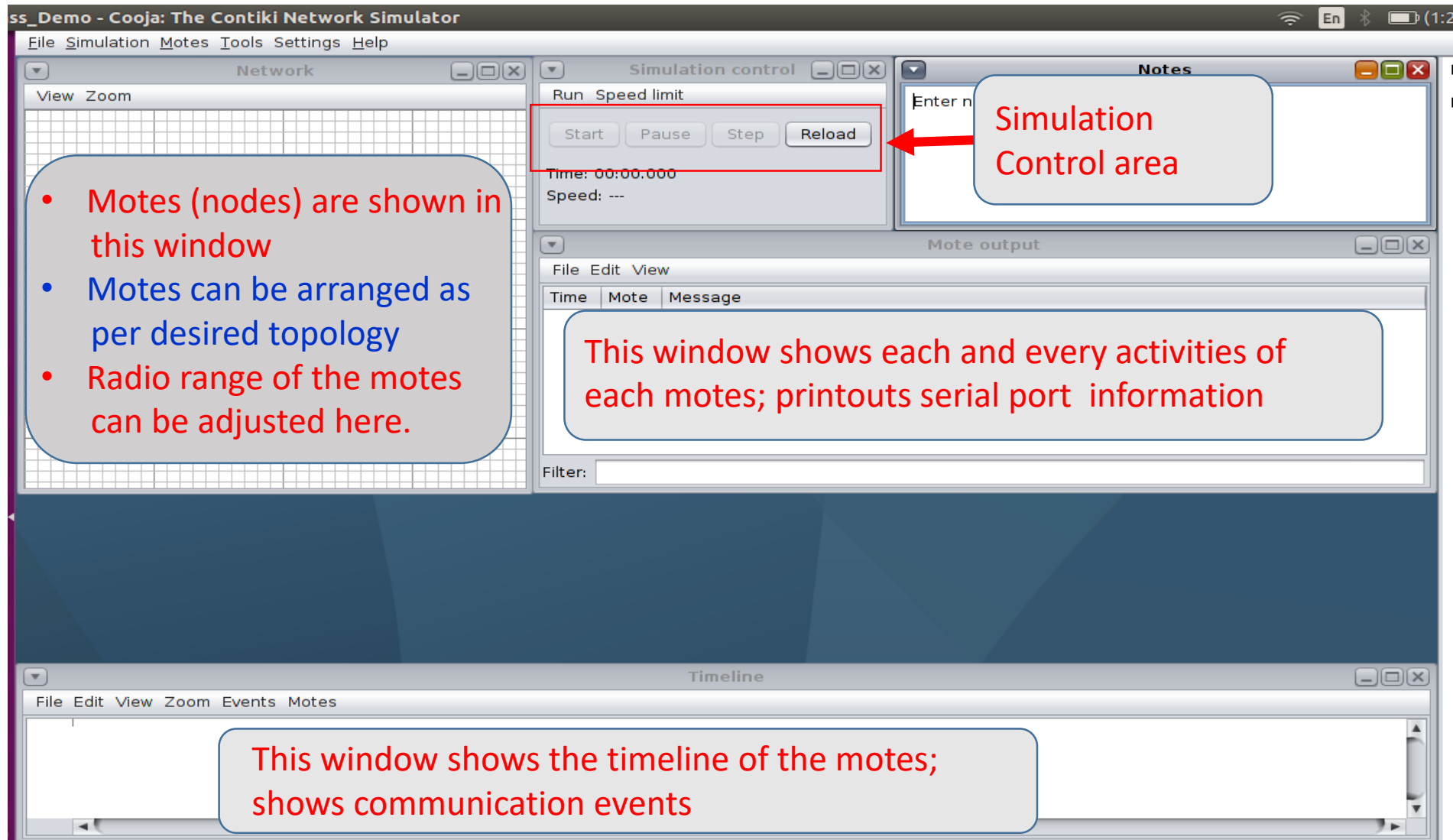
➤ First interface after the starting of the simulator



Create a new Simulation

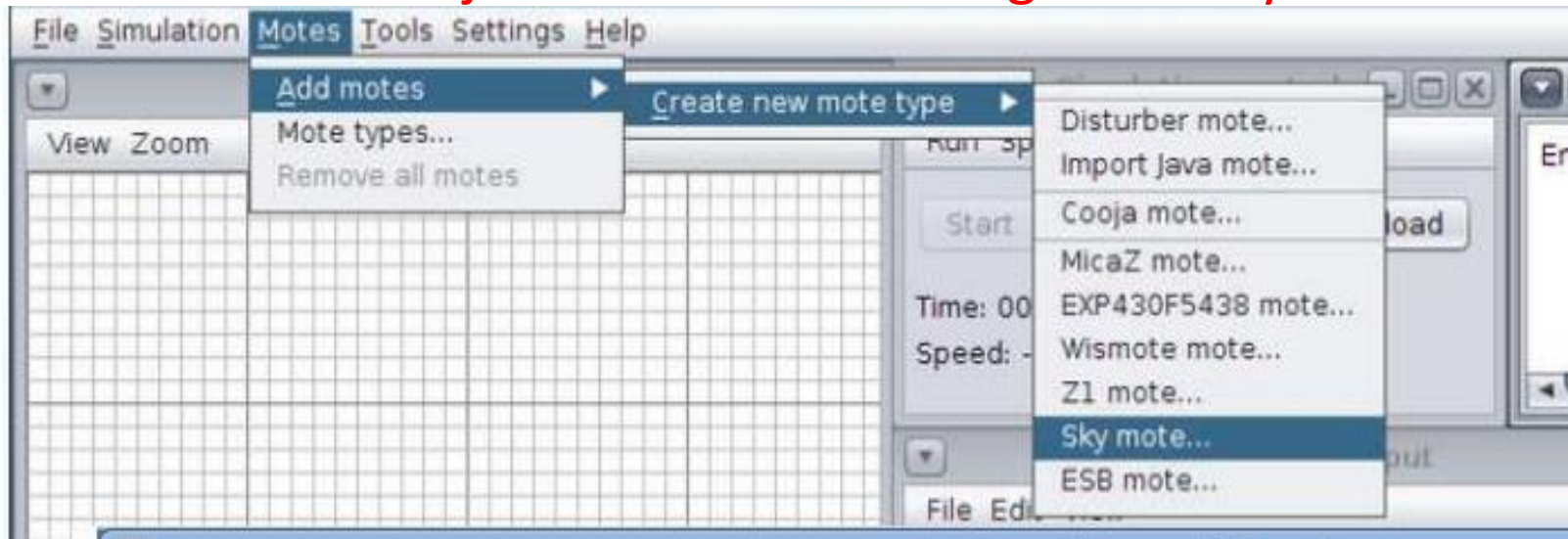


Basic Simulation Interface



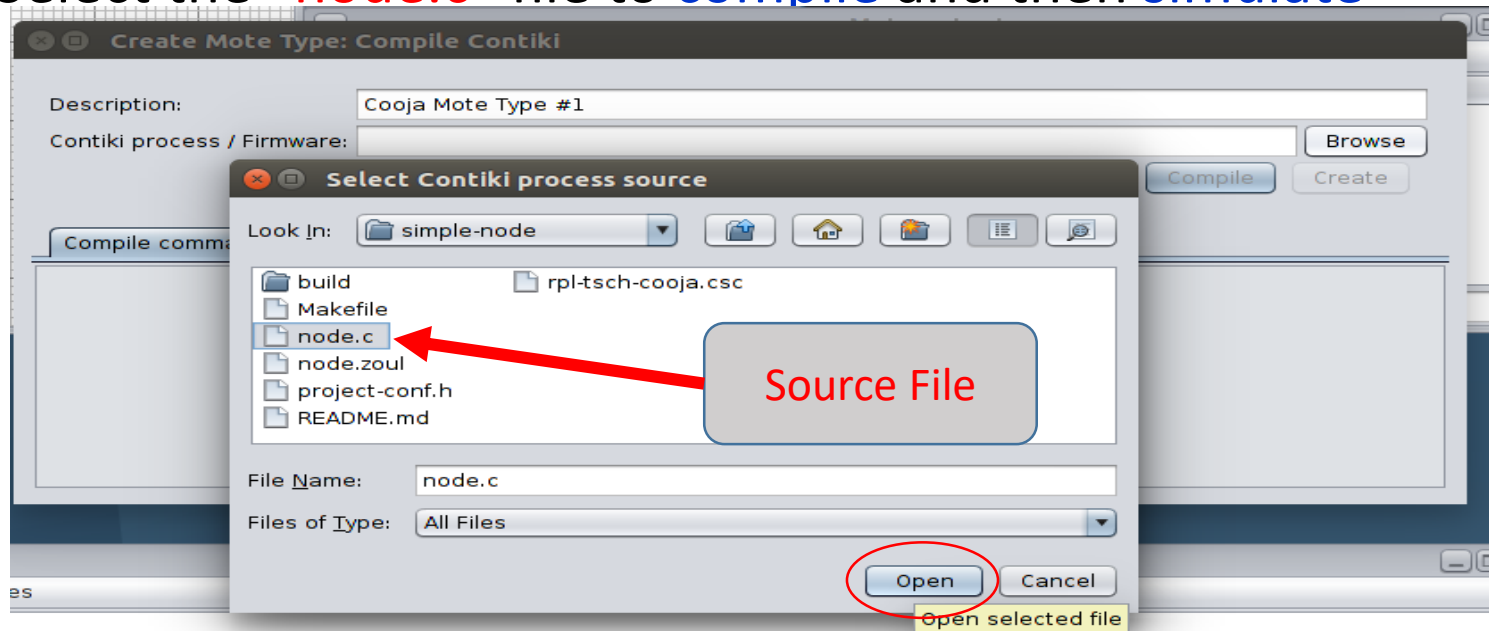
How to add motes (nodes)?

- Before simulation, **motes** need to be **added**
- In the “**Motes**” tab, click on “**Add motes**”
- Next, click on “**Create new mote type**” and select the desired available mote (e.g., **Cooja mote**, **sky mote**)
- Better to use “**Cooja**” mote on **low configuration system**



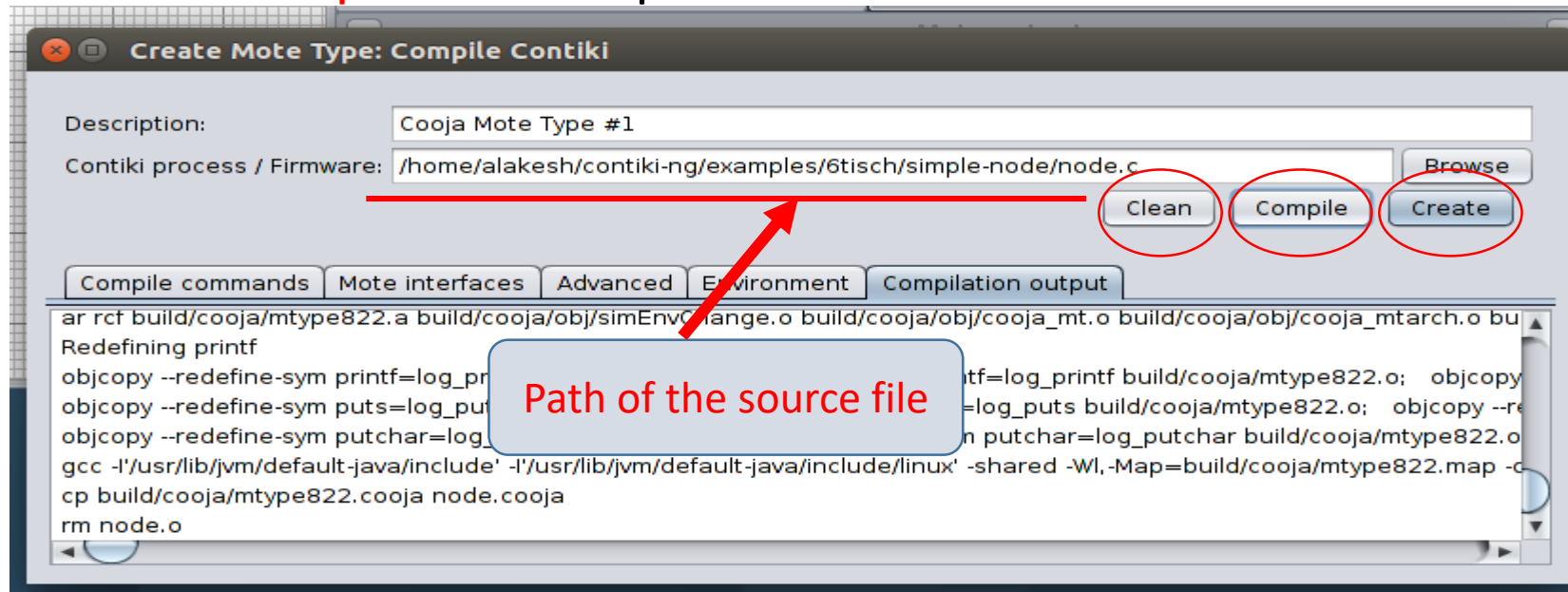
Select the source file

- Browse the source file that you want to simulate
- In the “example” folder of Contiki-NG, various source files are available
- Select the “node.c” file to compile and then simulate



Contd..

- Before Simulation, selected **source file** needs to be **compiled**.
- If any changes made in any of the source file, then click on **“clean”** before compiling, otherwise, no cleaning is required
- Click on **“compile”** to compile and then **“create”** to create the motes



Contd..

The screenshot displays the Cooja network simulator interface. The main window is divided into several panes: a 'Network' pane on the left with a grid view, a 'Simulation control' pane on the top right with buttons for 'Start', 'Pause', 'Step', and 'Reload', and a 'Mote output' pane on the bottom right. A dialog box titled 'Add nodes (Cooja Mote Type #1)' is open in the foreground. This dialog box contains the following fields and controls:

- 'Number of new motes': A text input field containing the value '25'.
- 'Positioning': A dropdown menu currently set to 'Linear positioning'.
- 'Position interval': A section with three rows of controls for X, Y, and Z coordinates. Each row has a text input field, a double-headed arrow, and another text input field.
 - X: Input field '0', arrow, input field '100'.
 - Y: Input field '0', arrow, input field '100'.
 - Z: Input field '0', arrow, input field '0'.
- At the bottom: Two buttons, 'Do not add motes' and 'Add motes'.

A red bracket is drawn on the left side of the 'Add nodes' dialog box, pointing to the list of instructions on the left.

- Add the number of nodes
- Select the position of nodes
 - Motes positions can be changed at any time during simulation
- Nodes position interval can be set

Interface during simulation-1

The screenshot displays the Contiki-NG simulation interface. The top menu bar includes File, Simulation, Motes, Tools, Settings, and Help. The main window is divided into several panels:

- Network:** A grid showing 25 motes (numbered 1-25) connected by blue lines representing network links.
- Simulation control:** Contains buttons for Run, Speed limit, Start, Pause, Step, and Reload. It also displays the current Time (00:07.642) and Speed (---).
- Notes:** A text area for entering notes.
- Mote output:** A window showing the output of the simulation. It includes a table with columns for Time, Mote, and Message. The output is filtered to show messages from Mote 24.
- Timeline showing 25 motes:** A horizontal timeline at the bottom showing the activity of 25 motes. A red circle highlights a specific event on the timeline for Mote 4.

Red arrows point from the labels "Motes activity" and "Event timeline" to the Mote output and Timeline windows, respectively.

Time	Mote	Message
00:00.035	ID:24	[INFO: Main] Starting Contiki-NG-release/v4.4-11-g9b2dfb0-dirty
00:00.035	ID:24	[INFO: Main] - Routing: RPL Lite
00:00.035	ID:24	[INFO: Main] - Net: sicslowpan
00:00.035	ID:24	[INFO: Main] - MAC: TSCH
00:00.035	ID:24	[INFO: Main] - 802.15.4 PANID: 0x81a5
00:00.035	ID:24	[INFO: Main] - 802.15.4 TSCH default hopping sequence length: 4
00:00.035	ID:24	[INFO: Main] Node ID: 24
00:00.035	ID:24	[INFO: Main] Link-layer address: 0018.0018.0018.0018

Interface during simulation-2

The screenshot displays the simulation interface with several key components:

- Network View:** A grid showing 25 nodes (numbered 1-25) connected by blue lines. A red callout box states: "Nodes can change their positions to anywhere".
- Simulation Control:** A panel with buttons for "Run", "Speed limit", "Start", "Pause", "Step", and "Reload". It also shows "Time: 00:17.335" and "Speed: ---".
- Notes Panel:** A section for entering notes.
- PowerTracker: 25 notes:** A table showing radio duty cycles for 25 nodes. The table has columns: "Mote", "Radio on ...", "Radio TX (...)", and "Radio RX ...". The row for "Contiki 13" is highlighted in red.
- Timeline:** A section at the bottom showing a timeline of events for 25 nodes.
- File Tab:** A callout box points to the "File" tab in the bottom panel, stating: "Saving option for nodes activities logfile under the 'File' tab".
- Radio Duty Cycles:** A callout box points to the "PowerTracker" table, stating: "Radio duty cycles of nodes in different micro-controller states Under 'Tool' tab, it is available".

PowerTracker: 25 notes Table:

Mote	Radio on ...	Radio TX (...)	Radio RX ...
Contiki 1	10.67%	0.33%	1.56%
Contiki 2	10.55%	0.13%	1.57%
Contiki 3	55.76%	0.07%	1.19%
Contiki 4	10.57%	0.16%	1.55%
Contiki 5	60.15%	0.00%	1.15%
Contiki 6	10.50%	0.13%	1.54%
Contiki 7	10.51%	0.16%	1.52%
Contiki 8	66.87%	0.13%	0.94%
Contiki 9	60.18%	0.00%	1.15%
Contiki 10	10.51%	0.14%	1.46%
Contiki 11	72.36%	0.03%	0.96%
Contiki 12	10.49%	0.18%	1.44%
Contiki 13	100.00%	0.00%	0.45%
Contiki 14	55.74%	0.03%	1.11%
Contiki 15	65.46%	0.00%	0.91%
Contiki 16	10.49%	0.12%	1.48%
Contiki 17	99.47%	0.00%	0.46%
Contiki 18	10.50%	0.19%	1.46%
Contiki 19	10.50%	0.10%	1.44%
Contiki 20	60.20%	0.00%	1.04%

FIT IoT-LAB



The Very Large Scale IoT Testbed

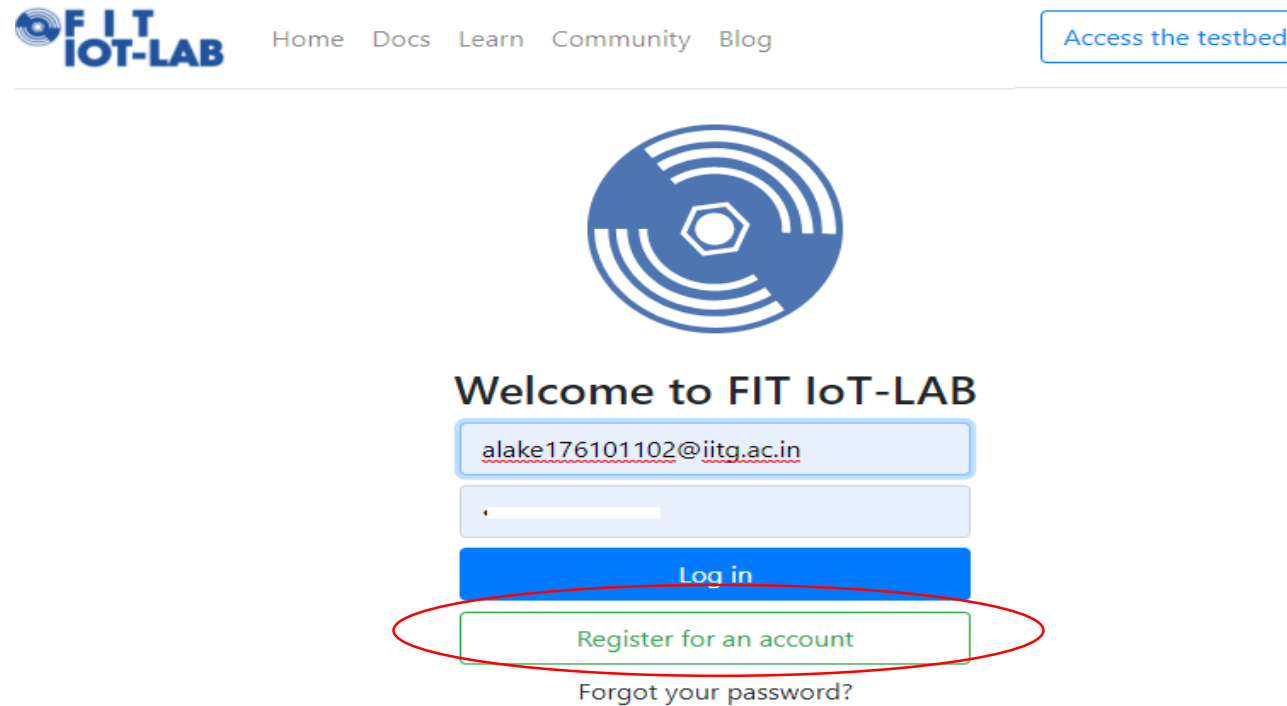
IoT-LAB provides a facility suitable for testing
networking with small wireless sensor devices and
heterogeneous communicating objects.

What is FIT IoT-LAB?

- FIT IoT-LAB is a **real open source IoT Testbed**, can be **accessible over Internet**
 - where, **real physical nodes** are deployed
 - provides a very large scale infrastructure for testing
 - small **wireless** sensor devices
 - and **heterogeneous** communicating objects.
- FIT IoT-LAB features over **1500 wireless sensor nodes** spread across **six** different **sites** in **France**.
- Nodes are either **fixed** or **mobile** and can be allocated in various topologies throughout all sites
- Node are with different **processor architectures** (**MSP430**, **STM32** and **Cortex-A8**) and different **wireless chips** (802.15.4 PHY @ **800 MHz** or **2.4 GHz**)

How to access FIT IoT-LAB?

- FIT IoT-LAB is available on <https://www.iot-lab.info/>
- To **access** the testbed, one should create his/her **account** in FIT IoT-LAB



The screenshot shows the FIT IoT-LAB website interface. At the top left is the FIT IoT-LAB logo. To its right are navigation links: Home, Docs, Learn, Community, and Blog. On the top right is a button labeled "Access the testbed". In the center is a large circular logo with a hexagon in the middle. Below this logo, the text "Welcome to FIT IoT-LAB" is displayed. Underneath is a login form with two input fields: the first contains the email address "alake176101102@iitg.ac.in" and the second is empty. Below the input fields are two buttons: a blue "Log in" button and a green "Register for an account" button. The "Register for an account" button is circled in red. Below the registration button is a link that says "Forgot your password?".

Testbed Status: after login

Home Docs Learn Community Blog

Running experiments (4)

Id	User	Elapsed	State	Ending	Nodes
227259	gundogan	47 minutes (7%)	Running	2020-08-27 07:13	(24 nodes) m3-188.grenoble, m3-189.grenoble, m3-193.grenoble, m3-195.grenoble,
227277	rehacek	1 day, 15 hours (4%)	Running	2020-10-05 20:33	(1 node) pycom-1.saclay
227430	grosso	2 hours, 46 minutes (69%)	Running	2020-08-26 21:15	(64 nodes) m3-1.strasbourg, m3-
227437	arashid	26 minutes (9%)	Running	2020-08-27 00:34	(10 nodes) samr21-1.saclay, samr21-10.saclay, samr21-11.saclay, samr21-12.saclay, s

Sites
6 sites: grenoble, lille, lyon, paris, saclay, strasbourg

Architectures
19 archis: a8 (at86rf231), arduino-zero (xbee), firefly (multi), frdm-kw41z (multi), lora-gw (sx1301), m3 (at86rf231), microbit (ble), nrf51dk (ble), nrf52832mdk (ble), nrf52840dk (multi), nrf52840mdk (multi), nrf52dk (ble), phynode (kw2xrf), pycom (fipy), rtl-sdr (none), samr21 (at86rf233), samr30 (at86rf212b), st-iotnode (multi), st-lrwan1 (sx1276)

Nodes
1417 nodes: 1079 available, 99 busy, 220 unavailable, 19 dead, 13 mobile

Nodes properties | **Testbed Activity**

Showing 1417 nodes | view on site map

Annotations:

- In this menu bar various options such as user experiments, new experiments, user resources etc., are available** (points to the top navigation bar)
- To create new experiment** (points to the '+' icon in the top navigation bar)
- Different location where the nodes are present** (points to the 'Sites' section)
- Architecture of nodes** (points to the 'Architectures' section)
- Status of available nodes in different location** (points to the 'Nodes' section)

How to perform an experiment?

- To **create and perform** a new experiment, user needs to **create new experiment**.

The screenshot shows the 'Schedule' form for creating a new experiment in the FIT IoT-LAB system. The interface includes a top navigation bar with the logo and links to Home, Docs, Learn, Community, and Blog. A toolbar on the right contains icons for a flask, a plus sign, a folder, a list, a user, and a refresh button. The main form is titled 'Schedule' and contains the following fields and options:

- Name:** A text input field containing 'CS578_IoT_Class'. An annotation points to this field with the text 'Name of the experiment'.
- Duration:** A numeric input field containing '20', followed by radio buttons for 'minutes' (selected) and 'hours'. An annotation points to the 'minutes' radio button with the text 'Duration of the experiment'.
- Start:** Two radio buttons: 'As soon as possible' (selected) and 'Scheduled'. An annotation points to the 'Scheduled' radio button with the text 'User can schedule experiments if nodes are not available at that moment'.
- Nodes:** A button with a left-pointing arrow and the text 'Nodes'. An annotation points to this button with the text 'Nodes need to be selected before submitting the experiment'.
- Submit experiment:** A blue button at the bottom of the form. An annotation points to this button with the text 'User can submit his/her experiment once the nodes are selected'.

Below the 'Nodes' button, there is a 'Summary' section that begins with the text 'Your experiment on 0 nodes is set to start as soon as possible...'. A yellow callout box at the top right of the form says 'To create new experiment' with an arrow pointing to the plus icon in the toolbar.

How to select nodes

- User can select nodes by,
 - Node properties, Host name/map and Node Id

The screenshot shows a web interface for selecting nodes. It is divided into two main sections. The top section, titled 'Nodes' with a count of 0, has a 'Select by' dropdown menu with three options: 'node properties' (circled in red), 'host name / map', and 'node id'. Below this is a prompt 'Select an architecture, site and quantity.' followed by three dropdown menus for 'Architecture', 'Site', and 'Qty', a checkbox for 'Mobile', and a green '+ Add to experiment' button. The bottom section also has a 'Select by' dropdown menu with the same three options. Below it is a prompt 'Pick nodes from the list. Add some filters or a search pattern.' followed by three filter dropdown menus: 'All Sites' (circled in red), 'All Architectures' (circled in red), and 'All Mobility' (circled in red). To the right of this section is another 'Nodes' panel with a 'Select by' dropdown menu where 'node id' is circled in red. Below this is a prompt 'Select a site, architecture and desired ids.' followed by three input fields: 'Site', 'Architecture', and 'IDs (e.g. 1-5+7)'. Three red arrows point from the filter dropdowns in the bottom section to three grey callout boxes at the bottom. The first box points to 'All Sites' and contains the text 'Select the sites such as grenoble, Lille, Lyon, etc.,'. The second box points to 'All Architectures' and contains the text 'Select the architecture such as M3, A8 etc.,'. The third box points to 'All Mobility' and contains the text 'Selects nodes mobilities i.e., mobile or not mobile'.

Select the sites such as grenoble, Lille, Lyon, etc.,

Select the architecture such as M3, A8 etc.,

Selects nodes mobilities i.e., mobile or not mobile

Contd..

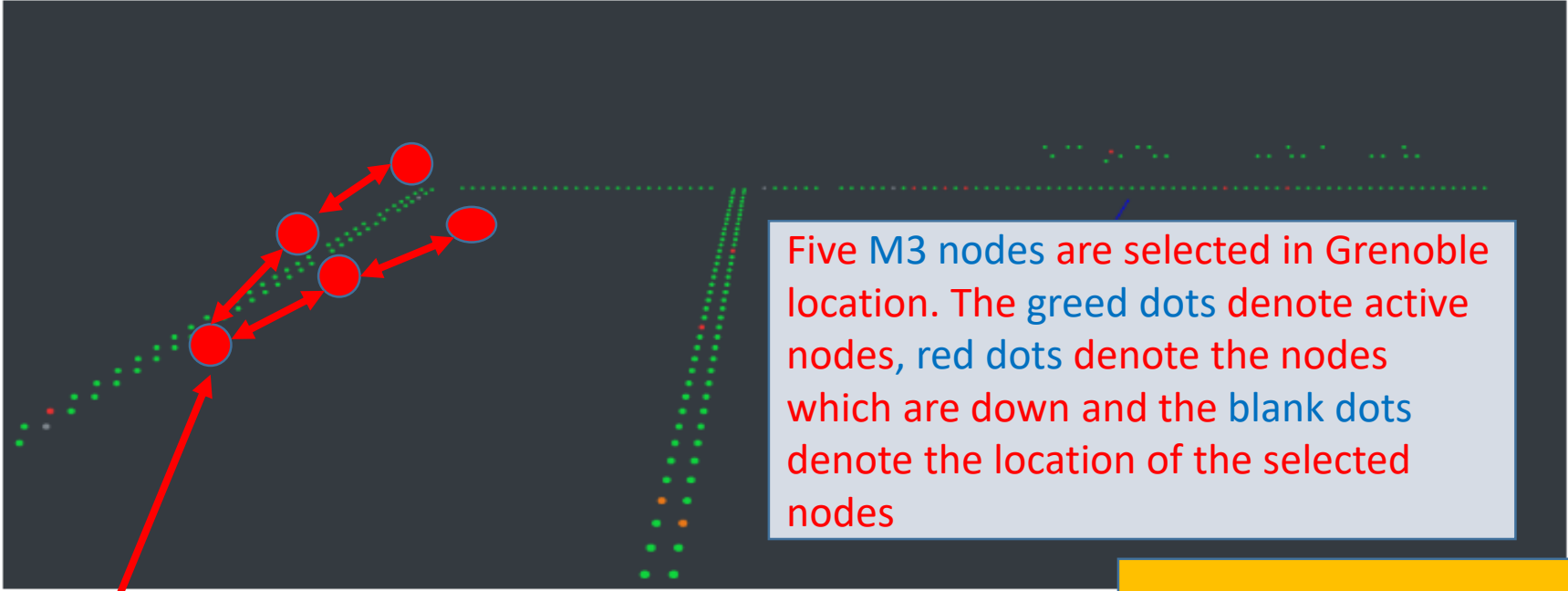
Select by [node properties](#) [host name / map](#) [node id](#)

Pick nodes from the list. Add some filters or a search pattern.

Filters [Grenoble](#) [M3 \(At86rf231\)](#) [All Mobility](#)

Search among 375 nodes [Add to experiment](#)

[View/select nodes on map](#)



Five M3 nodes are selected in Grenoble location. The green dots denote active nodes, red dots denote the nodes which are down and the blank dots denote the location of the selected nodes

5 nodes selected: [clear all](#)

[m3-142.grenoble=](#) [m3-124.grenoble=](#) [m3-121.grenoble=](#) [m3-108.grenoble=](#) [m3-109.grenoble=](#)

Here, the user firmware needs to be attached

Contd..

- Firmware is the binary file (.elf) generated using the contiki-NG.
- User can upload his/her modified code for example change in scheduling algorithm, change in RPL operation, change in TSCH operation etc.
- Depending on the type of selected node (M3, A8), the firmware varies.

To select a firmware

Assign a firmware (upload or select from list)

To add and store a firmware

Browse

+ New firmware

Identifier	Archi	OS	Description	
firefly-long-range	m3	contiki	wdwd	
node	m3			
nodeObj3	m3			

5 nodes selected: [clear all](#)

m3-142.grenoble× m3-124.grenoble× m3-121.grenoble× m3-108.grenoble× m3-109.grenoble×

Add firmware

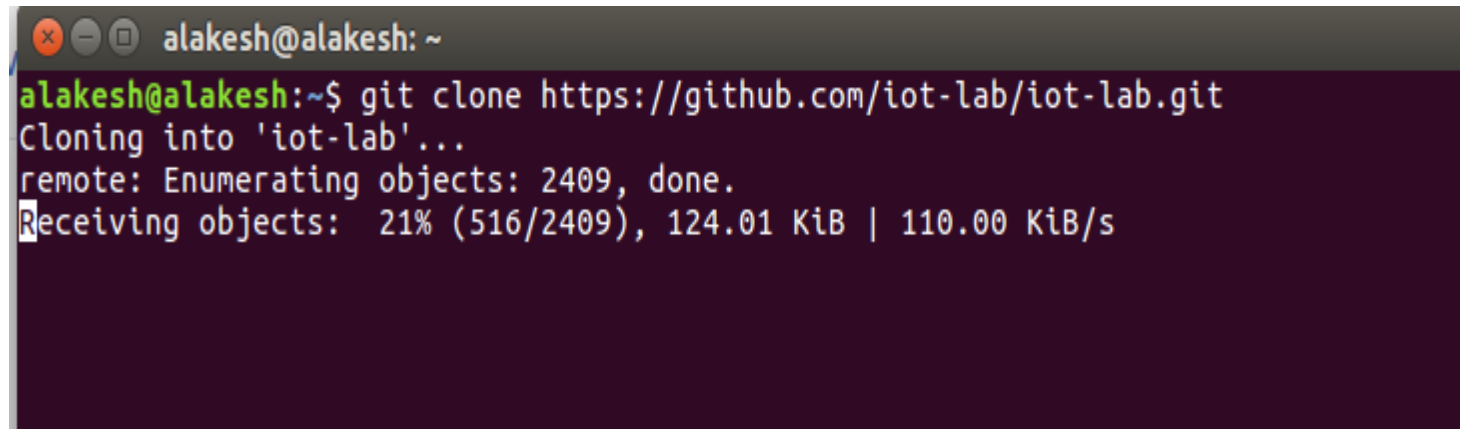
The screenshot displays the 'Presets' tab of the firmware management interface. A table lists three firmware options: 'firefly-long-range' (m3, contiki, wdwd), 'node' (m3), and 'nodeObj3' (m3). Each row has a download icon. Below the table, five nodes are listed as selected: m3-142.grenoble, m3-124.grenoble, m3-121.grenoble, m3-108.grenoble, and m3-109.grenoble. A 'clear all' link is next to the node list. At the bottom right, there is an 'Add firmware' button and a hardware icon.

How to generate a firmware

- To **compile** and **deploy firmware**, user can directly access the embedded boards by establishing **SSH connection**.
- The procedure to establish SSH connection can be found in
 - <https://www.iot-lab.info/docs/getting-started/ssh-access/>
- After establishing the SSH connection, following procedure needs to be followed for **firmware compilation**,
 - <https://www.iot-lab.info/legacy/tutorials/contiki-compilation/index.html>

Contd..

- However, it is **better to install the Contiki-NG** having the source code for the embedded boards that are available in **FIT IoT-LAB** on **user's own system**.
- Steps are as follows,
 - [git clone https://github.com/iot-lab/iot-lab.git](https://github.com/iot-lab/iot-lab.git)

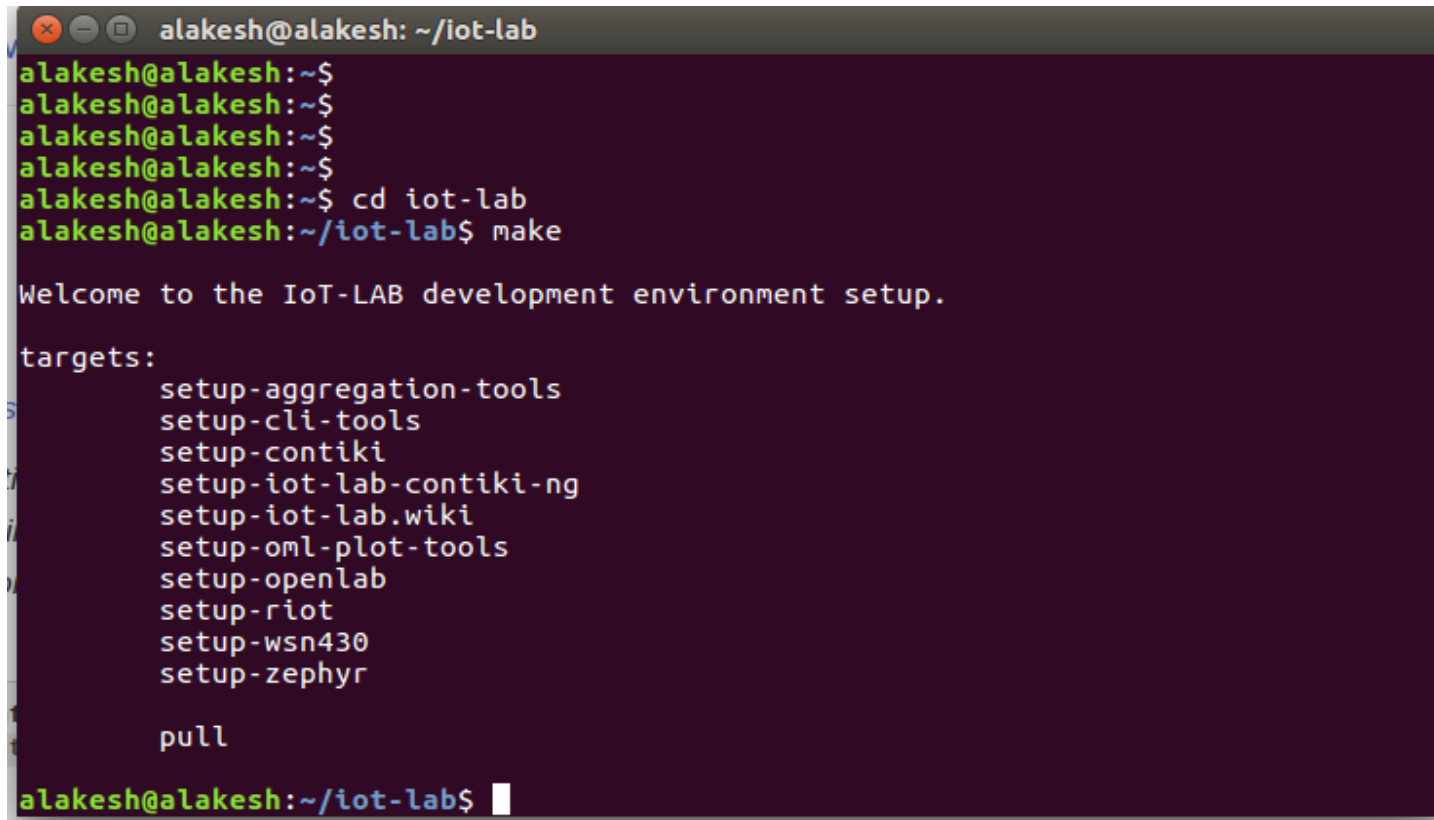
A terminal window with a dark background and light text. The window title is 'alakesh@alakesh: ~'. The prompt is 'alakesh@alakesh:~\$'. The command entered is 'git clone https://github.com/iot-lab/iot-lab.git'. The output shows 'Cloning into 'iot-lab'...', 'remote: Enumerating objects: 2409, done.', and 'Receiving objects: 21% (516/2409), 124.01 KiB | 110.00 KiB/s'.

```
alakesh@alakesh: ~  
alakesh@alakesh:~$ git clone https://github.com/iot-lab/iot-lab.git  
Cloning into 'iot-lab'...  
remote: Enumerating objects: 2409, done.  
Receiving objects: 21% (516/2409), 124.01 KiB | 110.00 KiB/s
```

Contd..

➤ *cd iot-lab*

➤ *make*

A terminal window with a dark purple background and white text. The window title is 'alakesh@alakesh: ~/iot-lab'. The terminal shows a sequence of commands: 'alakesh@alakesh:~\$' followed by four more instances of the same prompt. Then, 'alakesh@alakesh:~\$ cd iot-lab' is entered, and the prompt changes to 'alakesh@alakesh:~/iot-lab\$'. Finally, 'alakesh@alakesh:~/iot-lab\$ make' is entered. The output of 'make' is a list of targets: 'Welcome to the IoT-LAB development environment setup.', 'targets:', 'setup-aggregation-tools', 'setup-cli-tools', 'setup-contiki', 'setup-iot-lab-contiki-ng', 'setup-iot-lab.wiki', 'setup-oml-plot-tools', 'setup-openlab', 'setup-riot', 'setup-wsn430', 'setup-zephyr', and 'pull'. The terminal ends with the prompt 'alakesh@alakesh:~/iot-lab\$' and a cursor.

```
alakesh@alakesh: ~/iot-lab
alakesh@alakesh:~$
alakesh@alakesh:~$
alakesh@alakesh:~$
alakesh@alakesh:~$
alakesh@alakesh:~$ cd iot-lab
alakesh@alakesh:~/iot-lab$ make

Welcome to the IoT-LAB development environment setup.

targets:
    setup-aggregation-tools
    setup-cli-tools
    setup-contiki
    setup-iot-lab-contiki-ng
    setup-iot-lab.wiki
    setup-oml-plot-tools
    setup-openlab
    setup-riot
    setup-wsn430
    setup-zephyr

pull

alakesh@alakesh:~/iot-lab$
```

Contd..

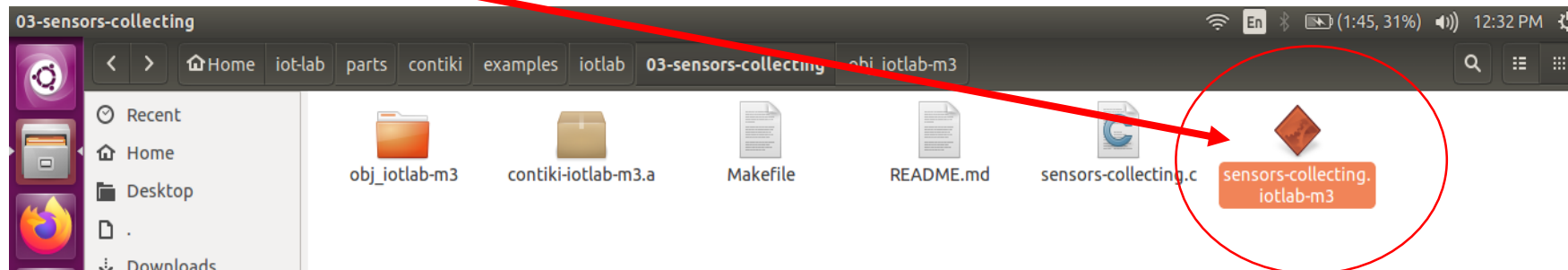
➤ *make setup- contiki*

➤ Go to the directory

➤ *cd parts/ contiki /examples/ iotlab /03-sensors-collecting*

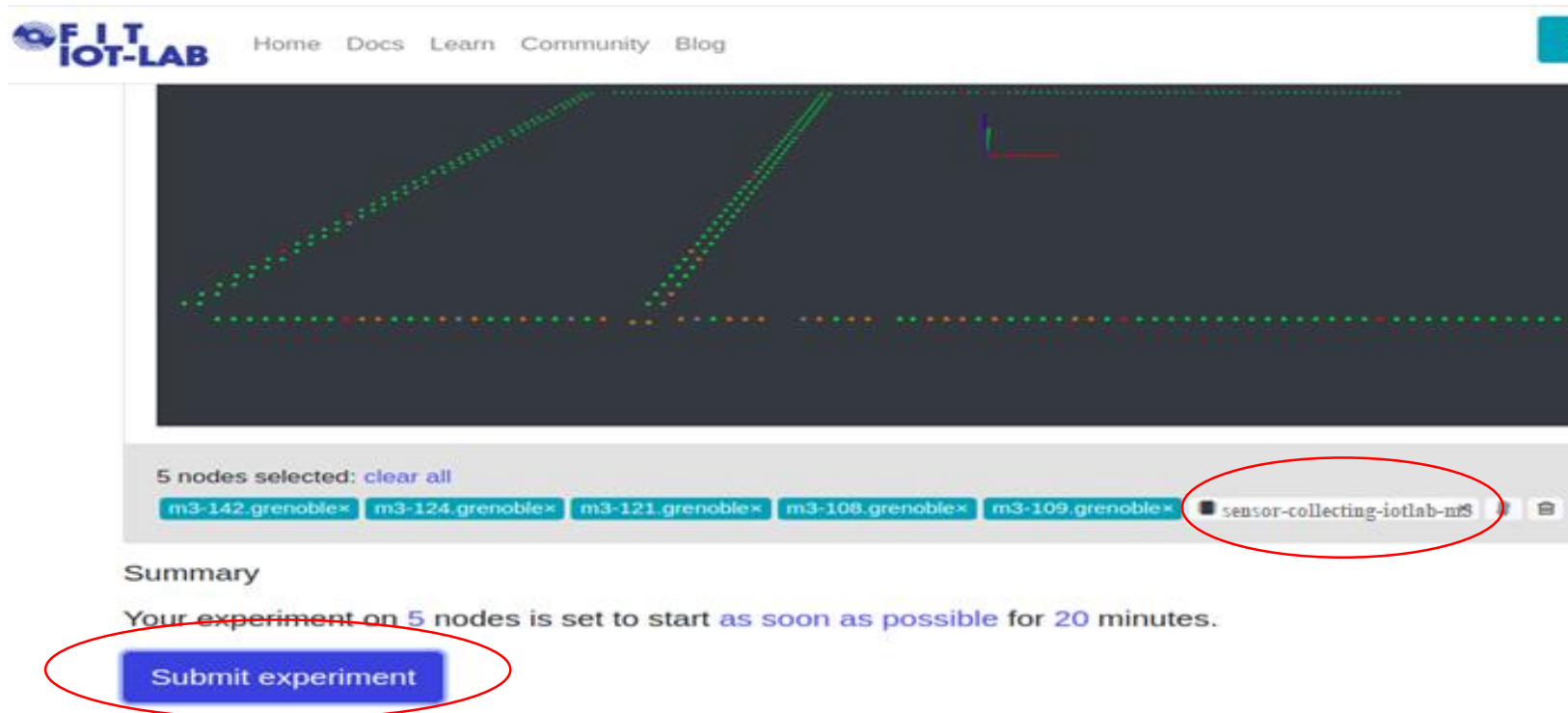
➤ *make TARGET=iotlab-m3* for m3 nodes and *make TARGET=iotlab-a8-m3* for a8 nodes

➤ One **.elf** binary file will be generated. This **firmware** is needed in FIT IoT-LAB



Upload a firmware

- Upload the **firmware** in FIT IoT-LAB
- After uploading, click on “**Submit experiment**” bottom to submit your experiment



Contd..

- Experiment is **launching**.

The screenshot shows the FIT IOT-LAB dashboard. At the top, there's a navigation bar with 'Home', 'Docs', 'Learn', 'Community', and 'Blog'. On the right, there are icons for a flask, a plus sign, a folder, a list, and a user profile. Below the navigation bar, there's a status summary: 0 Running, 1 Scheduled, and 119 Completed. The 'Scheduled' section is active, showing a table with one experiment.

Id	Name	Date	Duration	Nodes	State
227480	CS578_IoT_Class	2020-08-27 12:42	20 minutes	5	Launching

- **Firmware** can be **updated** during experiment also.

The screenshot shows the FIT IOT-LAB dashboard with a 'Select firmware' modal open. The modal has a search bar with 'sensors-collecting.iotlab-m3' and a 'Browse' button. Below the search bar, there are tabs for 'My firmwares' and 'Presets', and a '+ New firmware' button. A table lists available firmwares:

Identifier	Arch	OS	Description
firefly-long-range	m3	contiki	wdwd
node	m3		
nodeOBJ3	m3		

At the bottom of the modal, there are 'Close' and 'Flash' buttons. The background shows an experiment in progress with a 'Running' state and a list of nodes.

Flashing firmware

- Flashing a **firmware** during experiment

FIIT IOT-LAB Home Docs Learn Community Blog

Experiment CS578_IoT_Class #227482

User **kalita**
Submitted **2020-08-27 12:58:37**
Started **2020-08-27 12:58:38**
Duration **0 minutes (0%)** of **20 minutes**
Nodes **5**
State **Running**

Stop **Download**

Flashing firmware... **x**
firmware format elf **x**

Different actions can be performed during experiments

Actions on 5 nodes

Nodes	UID	Firmware	Monitoring	Deployment	Actions
m3-108.grenoble.iot-lab.info	8477	border-router.iotlab-m3		Success	▶ ⏻ ↺ 📶 🌡 >_ ✓
m3-109.grenoble.iot-lab.info	1062	border-router.iotlab-m3		Success	▶ ⏻ ↺ 📶 🌡 >_ ✓
m3-121.grenoble.iot-lab.info	8572	border-router.iotlab-m3		Success	▶ ⏻ ↺ 📶 🌡 >_ ✓
m3-124.grenoble.iot-lab.info	9772	border-router.iotlab-m3		Success	▶ ⏻ ↺ 📶 🌡 >_ ✓
m3-142.grenoble.iot-lab.info	a783	border-router.iotlab-m3		Success	▶ ⏻ ↺ 📶 🌡 >_ ✓

experimen....tar.gz ^

Show all **x**

Getting the output

- **Output** can be obtained from each nodes.

Experiment CS578_IoT_Class #227482

User **kalita**

Submitted **2020-08-27 12:58:37**

Started **2020-08-27 12:58:38**

Duration **2 minutes (10%)** of 20 minutes











Nodes **5**

State **Running**

Stop

Download

Actions on 5 nodes

Nodes	UID	Firmware	Monitoring	Deployment	Actions
m3-109.grenoble.iot-lab.info	8477	border-router.iotlab-m3		Success	    
<pre>accel: -15 -47 -1035 xyz mg magne: 389 49 -706 xyz mgauss light: 0.076294 lux press: 990.955078 mbar gyros: 183 -1426 -700 xyz m°/s accel: -16 -42 -1051 xyz mg magne: 392 50 -710 xyz mgauss light: 0.076294 lux press: 990.739990 mbar gyros: -236 -52 -525 xyz m°/s accel: -12 -47 -1035 xyz mg magne: 381 48 -710 xyz mgauss light: 0.076294 lux press: 990.780029 mbar gyros: 603 113 210 xyz m°/s magne: 392 47 -709 xyz mgauss accel: -14 -43 -1048 xyz mg light: 0.076294 lux press: 990.868896 mbar</pre>					
m3-109.grenoble.iot-lab.info	1062	border-router.iotlab-m3		Success	    
<pre>accel: -39 -15 -1047 xyz mg light: 0.076294 lux press: 990.563721 mbar magne: 162 272 177 xyz mgauss</pre>					

To check the output from each nodes

Output from the sensors attached to the node m3-108

Thank You

**For any query
contact: alakesh.kalita1025@gmail.com**