Tutorial 1: Hands on Cooja Simulator

AESE Group

January 19, 2018

1. Introduction

Contiki is an open source operating system for sensor networks and other networked embedded devices. It connects tiny low-cost, low-power microcontrollers to the Internet providing powerful low-consumption communication [1]. It supports fully IPv6 and IPv4 standards, along with the recent low-power wireless standards: 6LowPAN, RPL, CoAP. Instant Contiki is an entire Contiki development environment (i.e. an Ubuntu Linux installation) that runs in VMWare player [2]. It supports the development of Contiki applications which are written in standard C. Also, Contiki provides Cooja, a network simulator that is implemented in Java. Cooja allows developers to test their code and systems before burning these on the target hardware. Unlike most simulators Cooja gives the possibility of emulating real hardware platforms.

This tutorial will give you an insight into the Cooja simulator environment. We will use Cooja to set up a basic RPL-based network consisting of TMote Sky motes (nodes). You will learn how to create network layouts, compile motes and examine output.

2. Getting Started

Contiki environment has been installed in all the Lab's PCs, and for the purposes of this lab you can go directly to Section 2.2. However, if you wish to set up the environment on your local machine there are two options available: *Instant Contiki* that runs within the virtual machine environment (see Section 2.1) or *Contiki repository* that runs on the native machine (see Section 2.2).

2.1 Install Instant Contiki

The Instant Contiki development environment is an Ubuntu Linux installation that runs within the VMware Player virtual machine environment. If you wish to set up the Instant Contiki environment on your machine, please follow the instructions given on Contiki website (see *Step 1: Grab Instant Contiki*):

http://www.contiki-os.org/start.html

2.2 Clone Contiki Repository

If you wish to install Contiki on your native machine without Instant Contiki environment, you should follow the instructions below:

- Open a terminal window and go to your home directory:

```
cd ~
```

- Clone the Contiki repository from GitHub to your local folder:

```
git clone https://github.com/contiki-os/contiki.git
cd contiki
```

```
git checkout release-3-0
git submodule sync && git submodule update –init
```

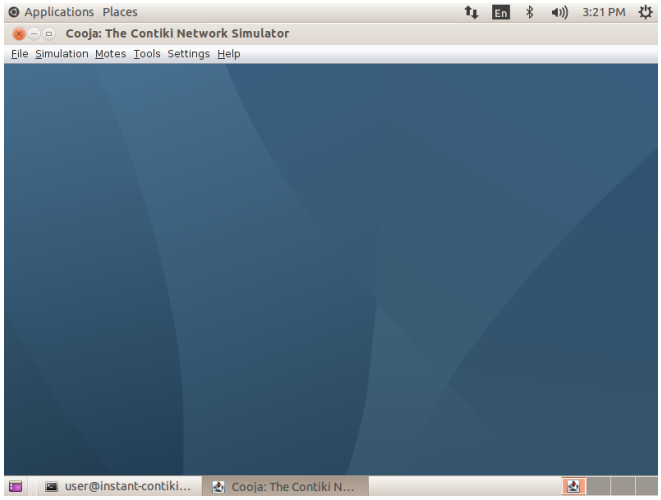
Also, the master branch on GitHub is the latest version; however, it is **NOT** stable. To run the Cooja coursework, you need to use the release 3.0 using the command above.

2.3 Start Cooja

To start the Cooja open a terminal window and use the following commands:

```
cd contiki/tools/cooja
ant run
```

It will take some time until Cooja is started, so please be patient. When Cooja is compiled, it will start with a following window:

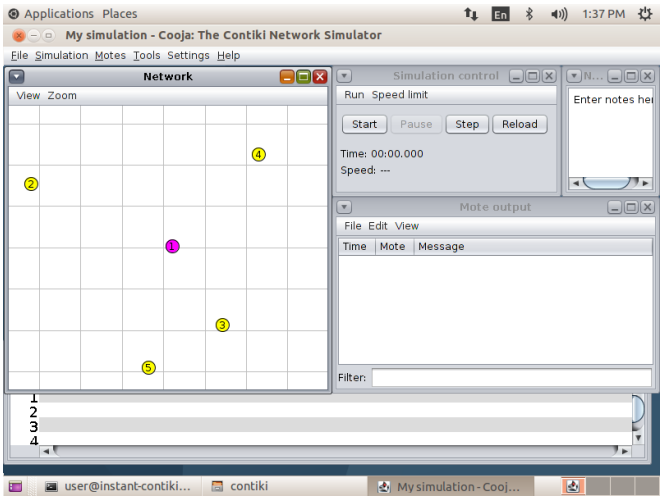


3. Run Simulation in Cooja

In this section, you will use a prebuilt simulation to familiarise yourself with the environment. The simulation file **sim1.csc** can be downloaded from CATE. The file should be saved to the contiki folder.

3.1 Open Simulation

From the **File** menu choose the option **Open simulation**, then **Browse** and choose the file **sim1.csc**. After opening the simulation file, you should get a following window:



3.2 Simulation Windows

- **Network window**, at the top left of the screen, shows all the motes in the simulated network. The network consists of one sink mote (ID1) and four sender motes (ID2-ID4). The **View** dropdown menu shows the various options that can be modified to display the network's factors.
- **Timeline window**, at the bottom of the screen, shows all communication events in the simulation over time. The events can be filtered using the dropdown menu and saved to a file if necessary.
- **Mote output window**, on the right side of the screen, shows all serial port printouts from all the motes (i.e. the output from the *printf* command). It can be very useful in the analysis of more complex networks' results and printouts can be exported to a file.
- **Notes window** on the top right is where you can put notes for your simulation.
- **Simulation control window** is where you start, pause, or reload your simulation. Different speed limits can be chosen.

3.3 Starting the Simulation

Click the **Start** button in the *Simulation control* window to start the simulation. Observe the simulation windows and different options available there. You can also **Pause**, **Stop** or **Reload** your simulation.

3.4 Exploring Additional Features

In the case of this demonstration the following options are already switched on: Mote IDs, Mote type, 10m background grid and Radio environment (UDGM). These can be found in the **View** dropdown menu of the Network Window. Some additional Cooja features that can be switched on/adjusted are:

- **Radio Traffic** – displays the exchange of messages between motes (to turn it on go to Network Window, View dropdown menu).
- **Transmission & Interference Range** – right click any mote to change these. The *green circle* represents the transmission range of the mote (i.e. the selected mote can communicate with all other motes within the green circle), while the *grey circle* around it represents its interference range (i.e. when the selected mote transmits packets, motes within the grey circle cannot receive packets correctly from other motes).

3.5 Add New Motes to Simulation

To add new mote, go to the **Motes** menu, where you click on **Add motes**. First step is to create a mote type by clicking on **Create new mote type** where you select one of the available mote types. In this simulation we are using TMote Sky mote type, so select **Sky mote**. Cooja will open *Create Mote Type* window where you can give a name to your mote type and choose you Contiki application. By using the **Browse** button go to the directory /home/user/contiki/examples/ipv6/rpl-collect and select the udp-sender.c source file. Click the **Compile** button. If the compilation is fine, click the **Create** button and enter the number of motes you wish to add. The process will be finished by using **Add motes** button.

3.6 Save the Simulation

To save the new network setup use menu item **File** and **Save simulation**. Make sure you use the .csc file extension.

4. Lab Task

1. Set up a WSN which consists of 20 motes by following these steps:
   - Radio medium: Unit Disk Graph Medium (UDMG): Distance Loss
   - Type of motes:
     - **One** Sky Mote running udp-server.c (from /home/user/contiki/examples/ipv6/rpl-udp folder) **randomly** positioned
     - **Nineteen** Sky Motes running udp-client.c (from home/user/contiki/examples/ipv6/rpl-udp folder) **randomly** positioned
   - Simulation area: 150m x 150m
   - Transmission (TX) range: 50m
   - Interference (INT) range: 70m

   All other parameters should be kept as default values. Make sure that you do not have isolated motes.

2. Run the simulation for 30 minutes with no speed limits and analyse *packet delivery ratio* (PDR) using Motes Output window. The PDR is a ratio between the total number of packets successfully received at the server side and the number of packets sent by the motes. In the ideal case, when all packets are successfully received, PDR equals 1 (or 100%). If no packets have been received, PDR equals 0 (or 0%).
3. Change the sink from the centre to the edge. What conclusion can be drawn from looking at the radio traffic?
4. Change the sending rate to 60 packets per minute and repeat Task 2. What conclusion can be drawn from the analysis?

5. References

[1] http://www.contiki-os.org/

[2] http://www.vmware.com/products/player/playerpro-evaluation.html