EXPERMINT: 01

- **Title**: IoT COOJA simulators

- **Aim:** To study and demonstrate use of IoT COOJA simulators on any real time application.

- **Theory:**

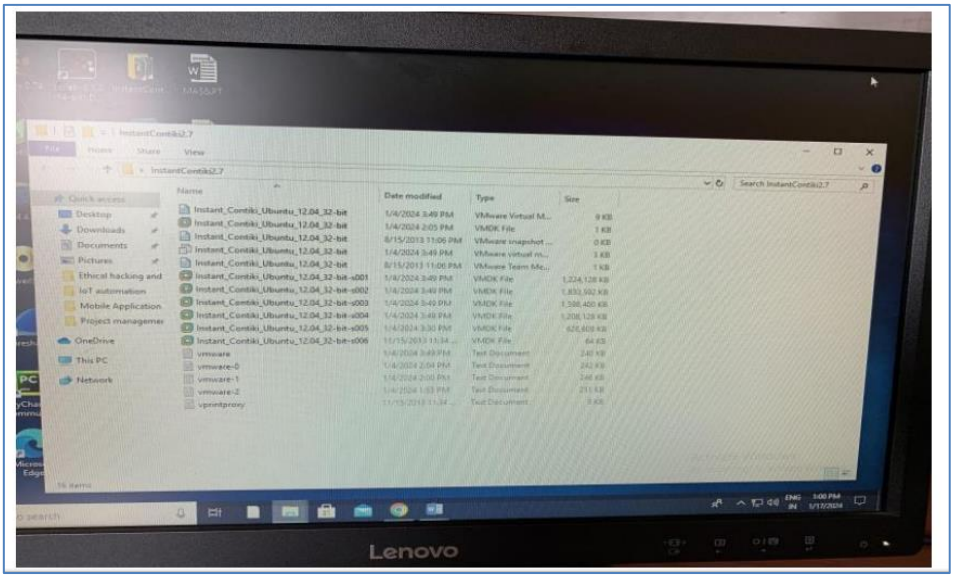
The Internet of Things (IoT) is the network of physical objects or things embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure. The IoT is enabled by the latest developments in RFID, smart sensors, communication technologies, and Internet protocols. Cooja Simulator is a network simulator specifically designed for Wireless Sensor Networks. A simulated Contiki Mote in COOJA is an actual compiled and executing Contiki system. The system is controlled and analyzed by COOJA. This is performed by compiling Contiki for the native platform as a shared library, and loading the library into Java using Java Native Interfaces (JNI). Several different Contiki libraries can be compiled and loaded in the same COOJA simulation, representing different kinds of sensor nodes (heterogeneous networks). COOJA controls and analyzes a Contiki system via a few functions. For instance, the simulator informs the Contiki system to handle an event, or fetches the entire Contiki system memory for analysis. This approach gives the simulator full control of simulated systems. Unfortunately, using JNI also has some annoying side-effects. The most significant is the dependency on external tools such as compilers and linkers and their run-time arguments. COOJA was originally developed for Cygwin/Windows and Linux platform, but has later been ported to MacOS. Java version 1.6 or later is required to run COOJA. We recommend using the latest version from Sun. In addition, the build tool ant is also required for building COOJA. Contiki is an open source operating system for the Internet of Things. Contiki connects tiny low-cost, low-power microcontrollers to the Internet. Contiki is a powerful toolbox for building complex wireless systems.

List of tools in Cooja and their Functionalities: ¬

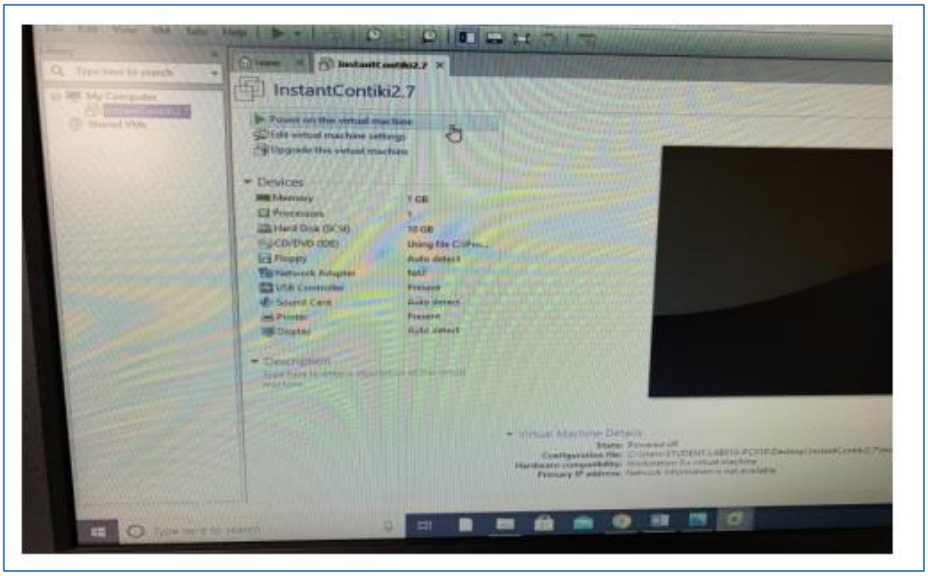
- Network – In the network tool window, the location of all wireless sensor nodes (WSN) aka nodes; can be seen. It is mainly meant for visualization purpose and to position the nodes. It will stay empty initially but once we start loading nodes we can see them in the network tool window. We can also see individual sensor transmission range (green circle) and detection range (grey area after green circle) by enabling the radio environment feature.
- Simulation Control – It is used to Start, Stop or Reload any simulation. It also shows the Speed index of the simulation. A rule of thumb is higher the number of nodes slower will the simulation run. (Note: Upon testing, I have often noticed with 100+ nodes simulation might freeze while running or after reload because memory is not properly cleaned by cooja in background.)
- Notes – Keep your notes or comments here. But, trust me, write your notes in the diary too
- Mote output – Shows every minute detail of Mote (Yup! Nodes are also called Motes). Search each node by typing in search bar ID: say for viewing node no. 11 data type ID:11 and hit enter.
- ¬ Timeline – Timeline of simulations where message and event logs are kept.
- ¬ Breakpoints – Used to set up breakpoints in simulations to debug issues. ¬
- Radio messages – Log of radio messages and packets as they are generated. It can be saved as .pcap file so that we can debug the messages in Wireshark later.
- ¬ Script editor – Developers can add their own custom scripts which will interact or collect data from Motes and give advance analytics data of the simulations.
- ¬ Mote duty cycle (Powertrace) – Used to check the Mote-wise sleep cycle percentage (%) separated by transmission(Tx) time and Reception(Rx) Time and their average values.

●**Procedure**:

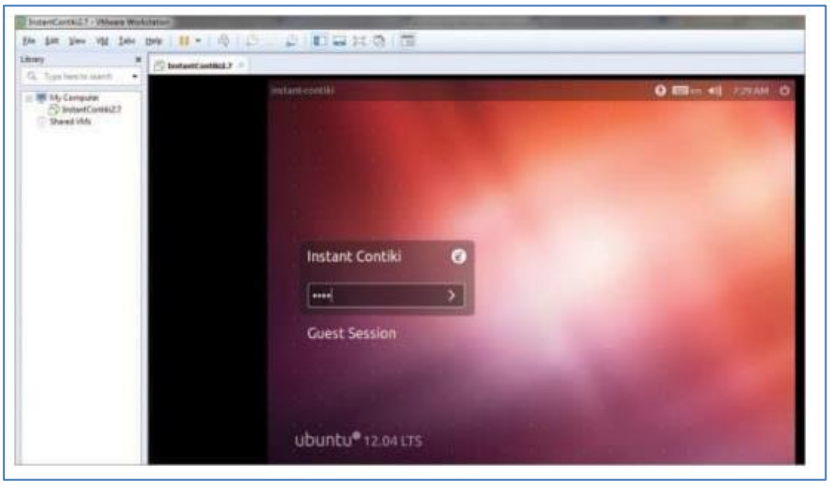
1) Download Instant Contiki and VMware



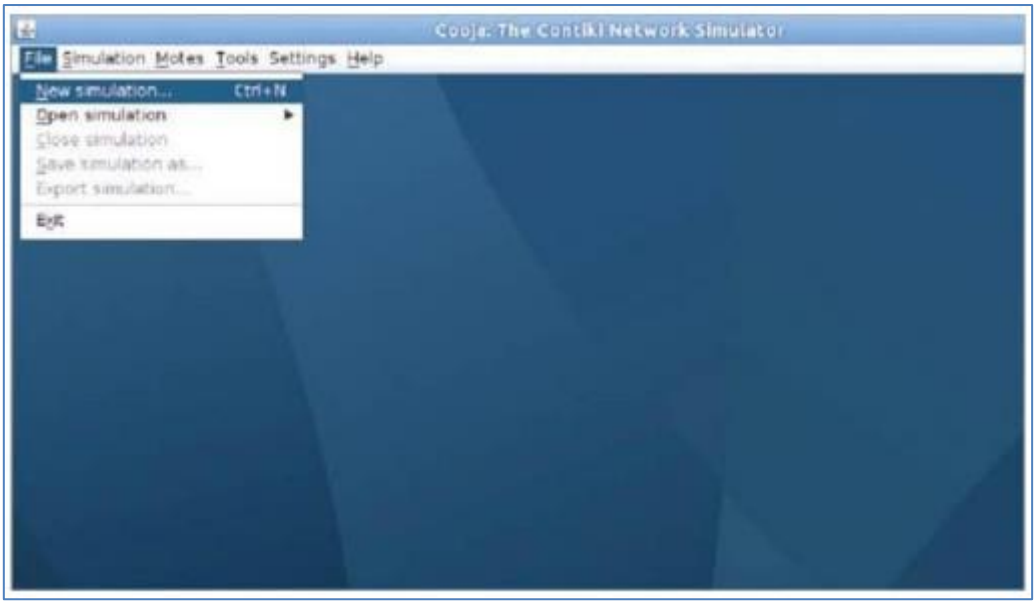
2) Open instant_ Contiki in VMware and click on Power on this virtual machine.



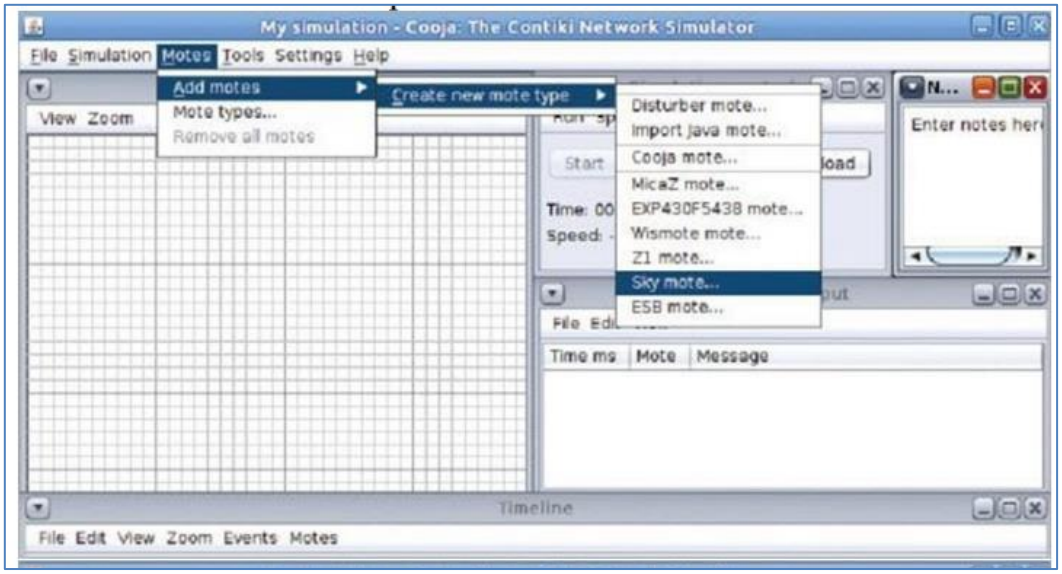
3) Window opened and ask to write password. Password is User. Then open Cooja simulator window.

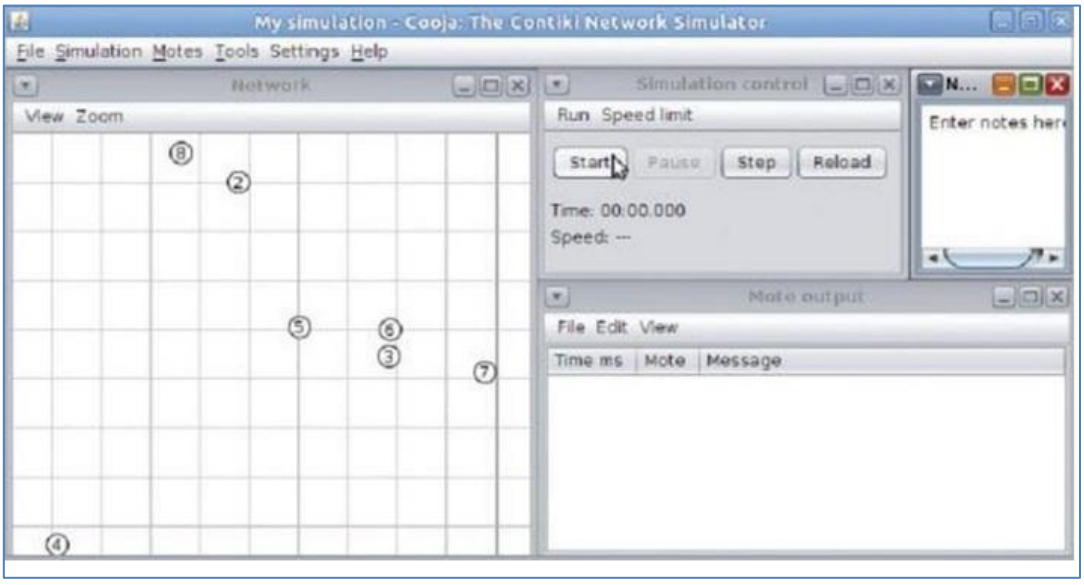


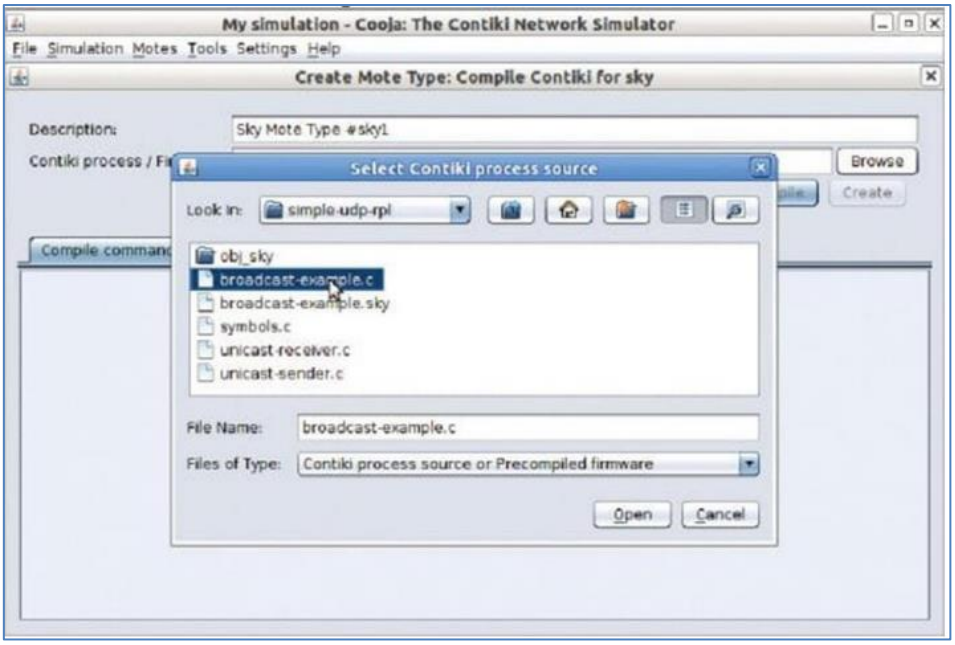
4) Start a new simulation.

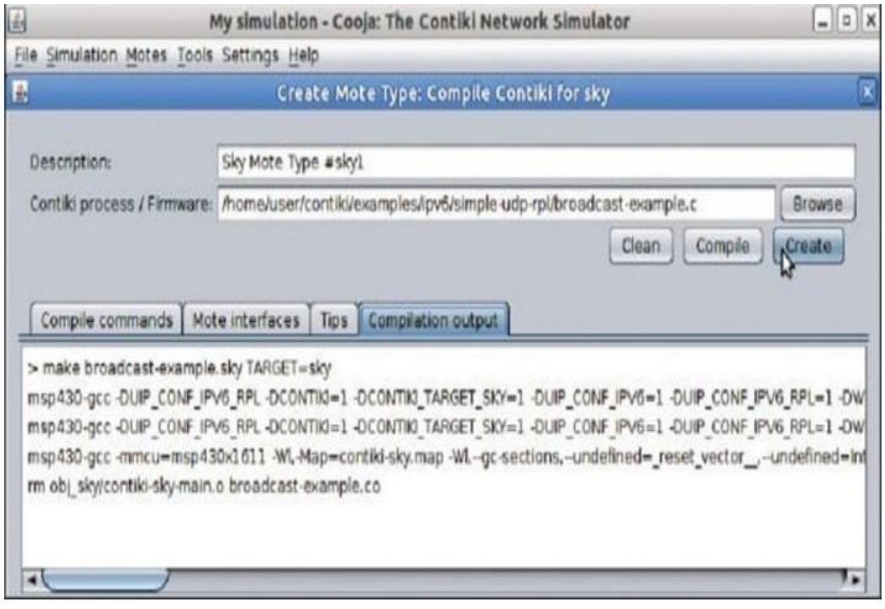


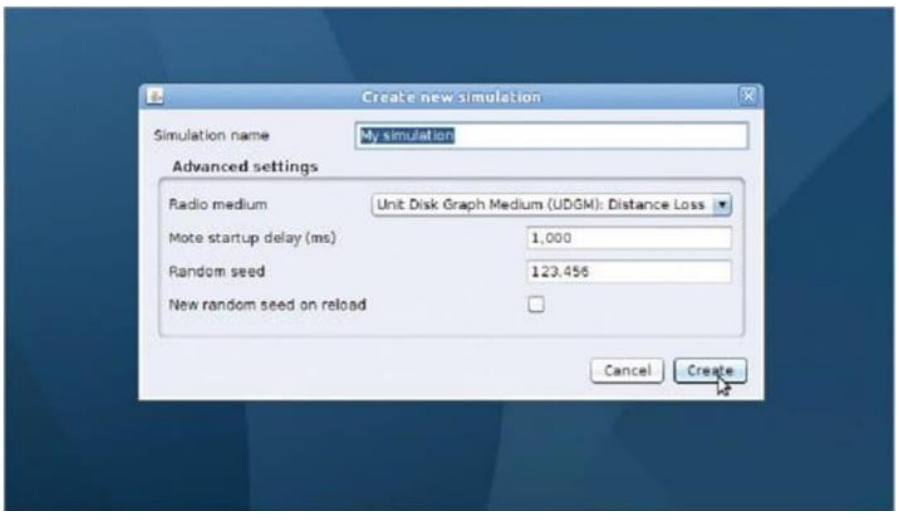
5) Add Motes and make the connection.



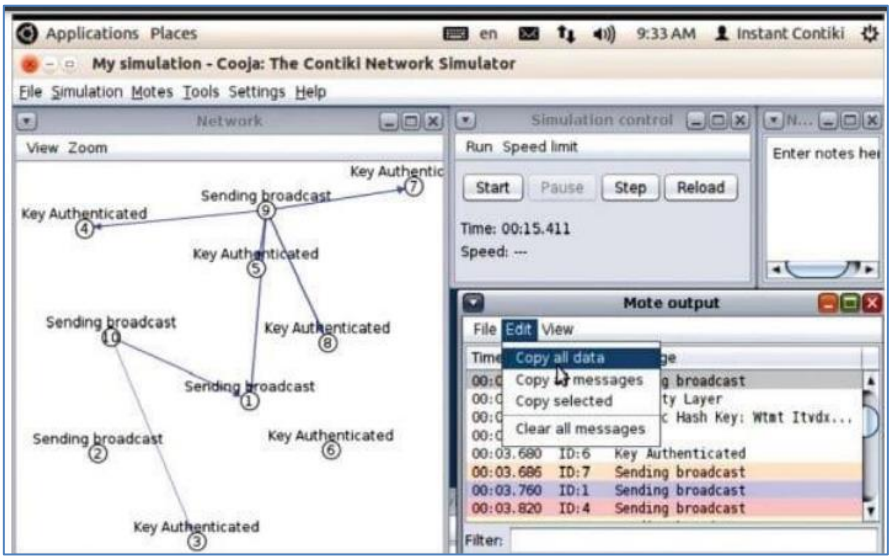



My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create Mote Type: Compile Contiki for sky
Description:
Sky Mote Type #sky1
Contiki process / Fi
Browse
Create
Select Contiki process source
Look in:
simple-udp-rpl
obj_sky
broadcast-example.c
broadcast-example.sky
symbols.c
unicast-receiver.c
unicast-sender.c
File Name:
broadcast-example.c
Files of Type:
Contiki process source or Precompiled firmware
Open
Cancel


My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create Mote Type: Compile Contiki for sky
Description:
Sky Mote Type #sky1
Contiki process / Firmware:
/home/user/contiki/examples/ipv6/simple-udp-rpl/broadcast-example.c
Browse
Clean
Compile
Create
Compile commands
Mote interfaces
Tips
Compilation output
> make broadcast-example.sky TARGET=sky
msp430-gcc -DUIP_CONF_IPV6_RPL -DCONTIKI=1 -DCONTIKI_TARGET_SKY=1 -DUIP_CONF_IPV6=1 -DUIP_CONF_IPV6_RPL=1 -DW
msp430-gcc -DUIP_CONF_IPV6_RPL -DCONTIKI=1 -DCONTIKI_TARGET_SKY=1 -DUIP_CONF_IPV6=1 -DUIP_CONF_IPV6_RPL=1 -DW
msp430-gcc -mmcu=msp430x1611 -Wl,-Map=contiki-sky.map -Wl,--gc-sections,--undefined=_reset_vector__,--undefined=Int
rm obj_sky/contiki-sky-main.o broadcast-example.co


Create new simulation
Simulation name
My simulation
Advanced settings
Radio medium
Unit Disk Graph Medium (UDGM): Distance Loss
Mote startup delay (ms)
1,000
Random seed
123.456
New random seed on reload
Cancel
Create

6) Connection Established.



●Benefits of Cooja simulator:

It allows the simulation of different levels from physical to application layer, and also allows the emulation of the hardware of a set of sensor nodes. Wireless Sensor Networks (WSNs) can be deployed using available hardware and software. The Contiki is an operative system compatible with a wide range of WSN hardware.

Major Advantages in Cooja IOT Simulator

- Communicate with the Nodes from the Host Machine
- It is used to connect the host machines and simulated nodes along with the web interface
- Capture Network Packets and View in WireShark
- The exploration of mesh network leads to capture the packets in the nodes and settle them as pcap file then it is analyzed by the Wireshark
- Timeline View of Events
- The timeline takes place in all the selected nodes and it is verified by the internal LEDs blink
- Emulation of 10 Metre Grid of Physical Environment
- The characterizes are placed in the nodes of wireless signal strength autonomously when the simulation takes place in the physical environment at 10-meter grid.

● Conclusion:

The text emphasizes the significance of using IoT COOJA simulators for studying and demonstrating real-time applications.