Perform CoAP Demo in Contiki

Aim : Perform CoAP Demo in Contiki

Hardware Required: Laptop/Computer

Software Required: Contiki Operating System, Oracle VM VirtualBox Manager

Knowledge Required: Basics of Operating Cooja Simulator in Instant Contiki OS

Theory and Steps to perform the task:

Contiki OS:

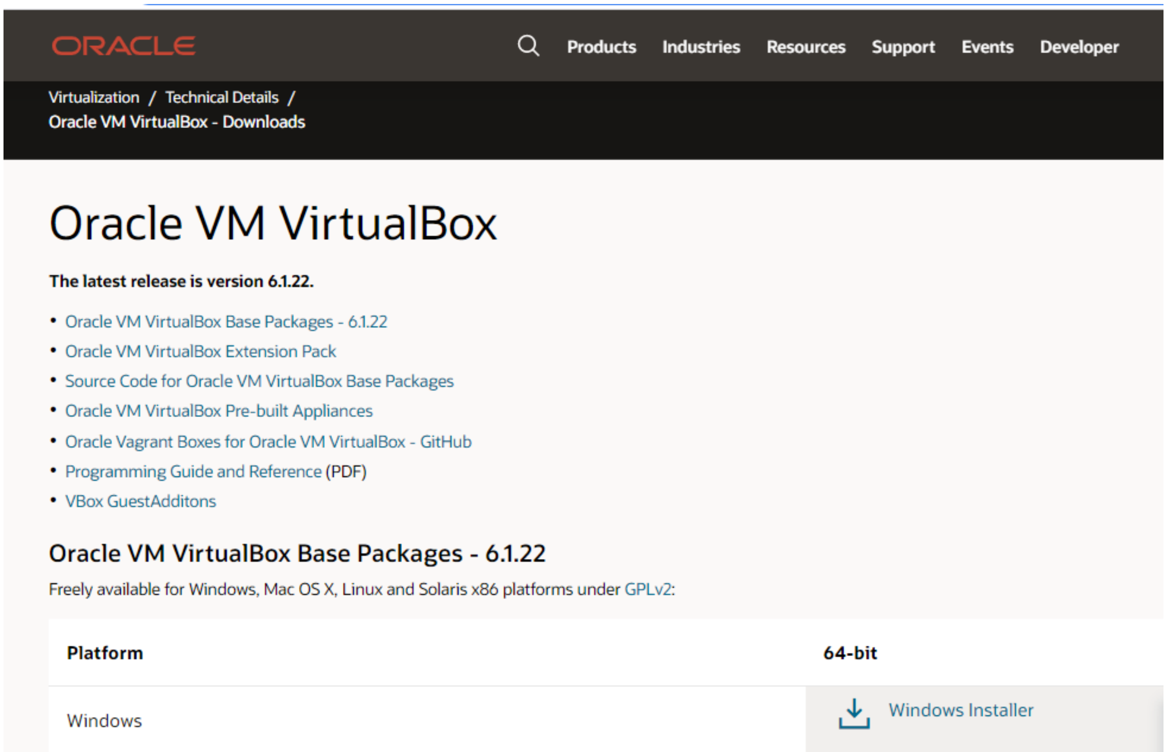
Contiki is an open source operating system that runs on tiny low-power microcontrollers and makes it possible to develop applications that make efficient use of the hardware while providing standardized low-power wireless communication for a range of hardware platforms.

Contiki is used in numerous commercial and non-commercial systems, such as city sound monitoring, street lights, networked electrical power meters, industrial monitoring, radiation monitoring, construction site monitoring, alarm systems, remote house monitoring, and so on.

Cooja:

A cross-layer java-based wireless sensor network simulator distributed with Contiki. 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.

First step is to download and install Oracle VM VirtualBox Manager link to download: https://www.oracle.com/in/virtualization/technologies/vm/downloads/virtualbox-downloads.html and then the next step is to download and install Contiki Operating System link to download: https://sourceforge.net/projects/contiki/



Virtualization / Technical Details /
Oracle VM VirtualBox - Downloads

Oracle VM VirtualBox

The latest release is version 6.1.22.

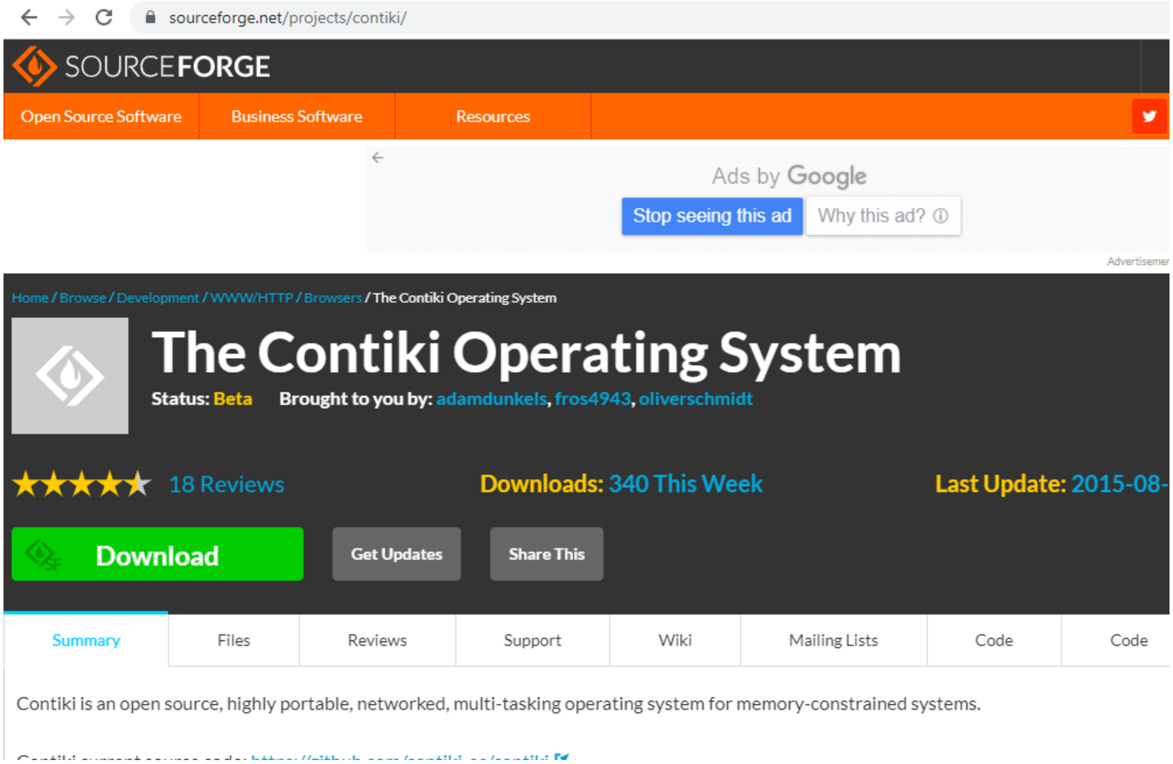
- Oracle VM VirtualBox Base Packages - 6.1.22
- Oracle VM VirtualBox Extension Pack
- Source Code for Oracle VM VirtualBox Base Packages
- Oracle VM VirtualBox Pre-built Appliances
- Oracle Vagrant Boxes for Oracle VM VirtualBox - GitHub
- Programming Guide and Reference (PDF)
- VBox GuestAdditons

Oracle VM VirtualBox Base Packages - 6.1.22

Freely available for Windows, Mac OS X, Linux and Solaris x86 platforms under GPLv2:

Platform	64-bit
Windows	Windows Installer

Figure 1.1 Downloading Oracle VM VirtualBox Manager



Home / Browse / Development / WWW/HTTP / Browsers / The Contiki Operating System

The Contiki Operating System

Status: Beta **Brought to you by:** adamdunkels, fros4943, oliverschmidt

★★★★★ 18 Reviews **Downloads:** 340 This Week **Last Update:** 2015-08-

Contiki is an open source, highly portable, networked, multi-tasking operating system for memory-constrained systems.

Contiki current source code: https://github.com/contiki-os/contiki

Figure 1.2 Downloading Contiki Operating System

After installation of Contiki in the VirtualBox start the Instant Contiki OS and in the login enter the password as ‘user’ and then login to the Contiki OS and then it is directed to the desktop:

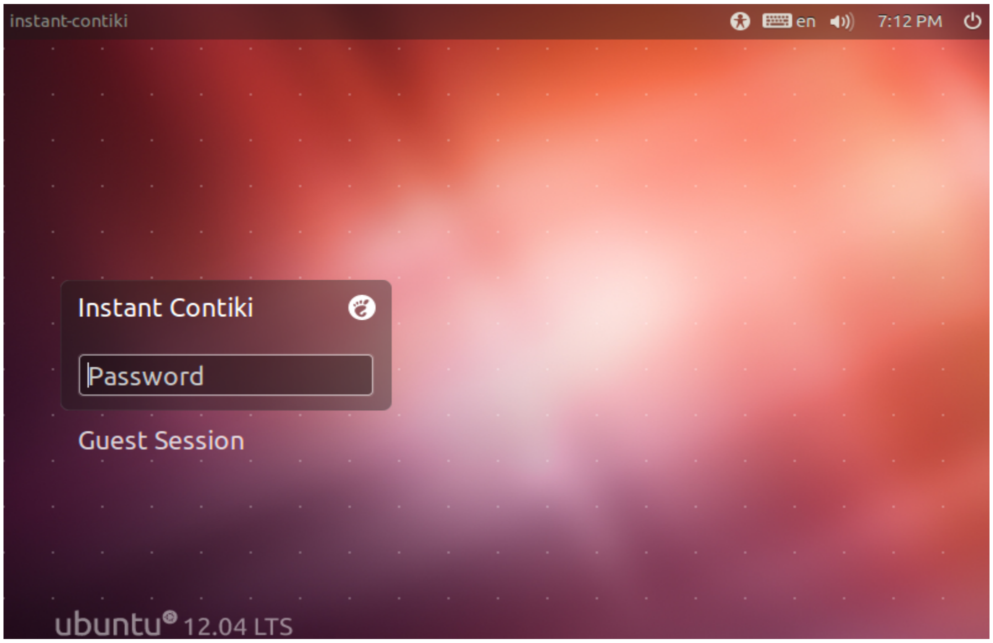


Figure 1.3 Login for Instant Contiki

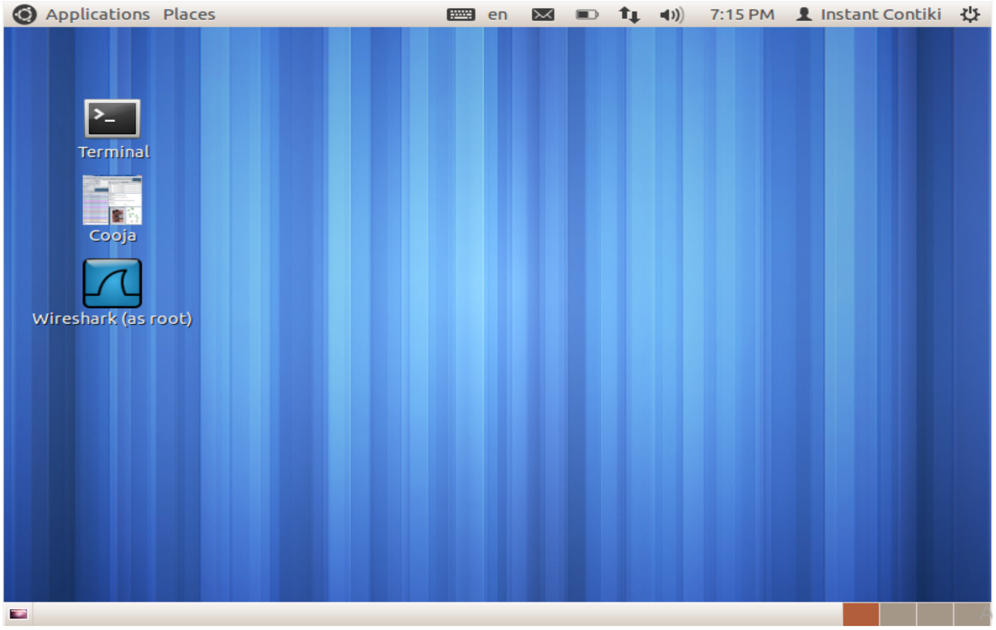
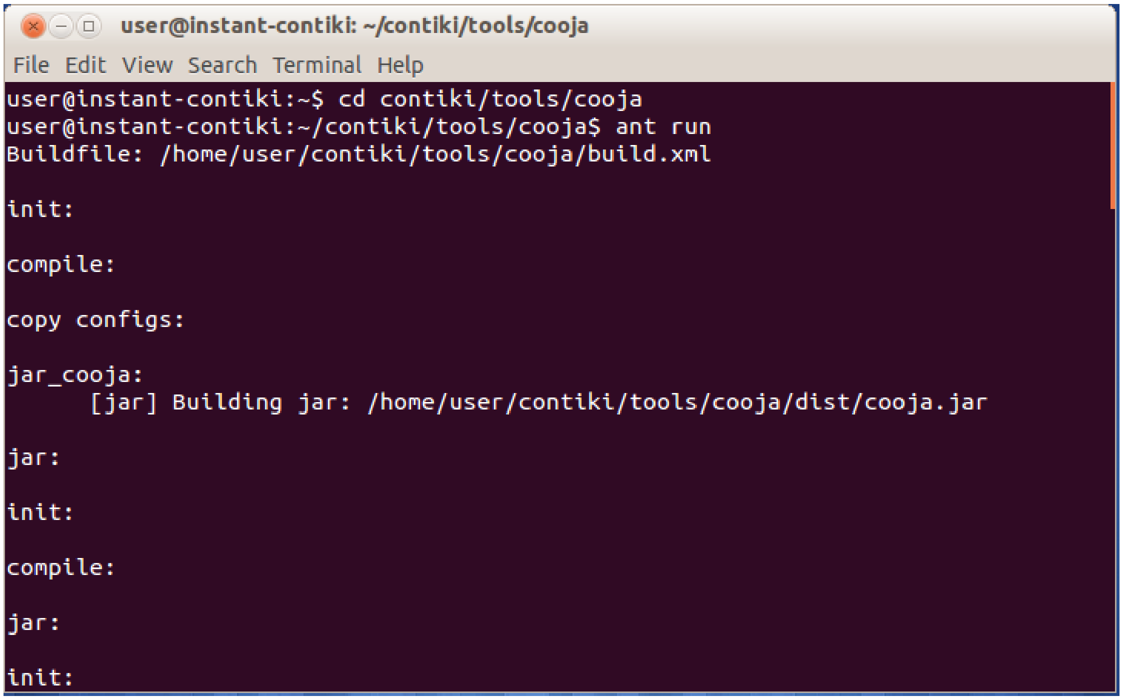


Figure 1.4 Instant Contiki Desktop

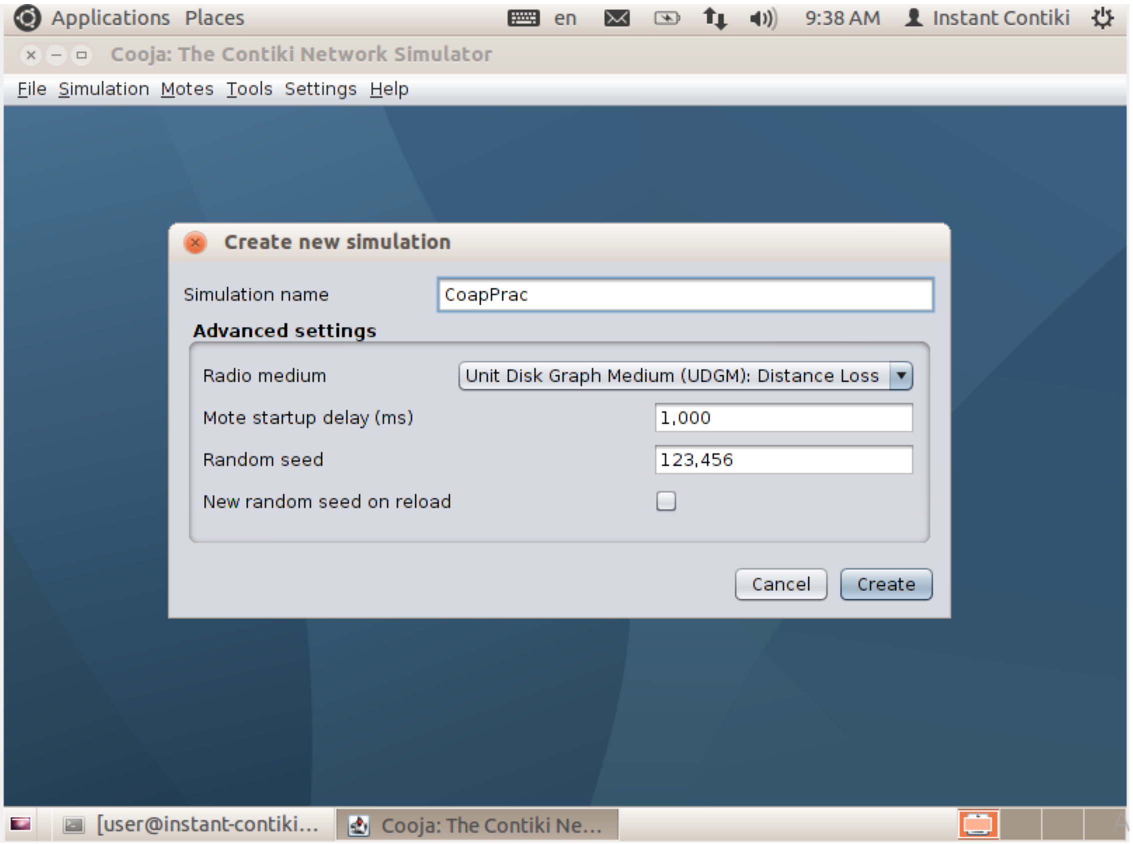
Launch the Cooja from desktop ,this will open the terminal and the Cooja Simulator:


user@instant-contiki: ~/contiki/tools/cooja
File Edit View Search Terminal Help

```
user@instant-contiki:~$ cd contiki/tools/cooja
user@instant-contiki:~/contiki/tools/cooja$ ant run
Buildfile: /home/user/contiki/tools/cooja/build.xml

init:

compile:

copy configs:

jar_cooja:
      [jar] Building jar: /home/user/contiki/tools/cooja/dist/cooja.jar

jar:

init:

compile:

jar:

init:
```

Figure 1.5 Running Cooja

Then Create a New Simulation and name it CoapPrac and then click on Create as in the figure


Applications Places
en 9:38 AM Instant Contiki
Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create new simulation

Simulation name: CoapPrac

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
[user@instant-contiki...
Cooja: The Contiki Ne...

Figure 1.6 New Simulation

Now Click on Motes in the Simulation and then add motes and then choose the sky mote :

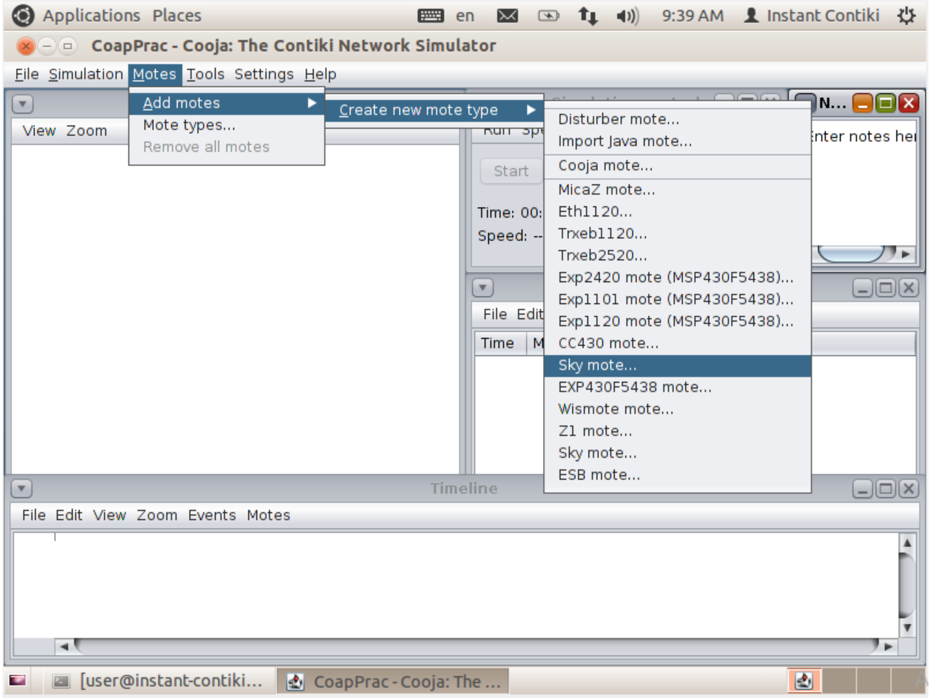
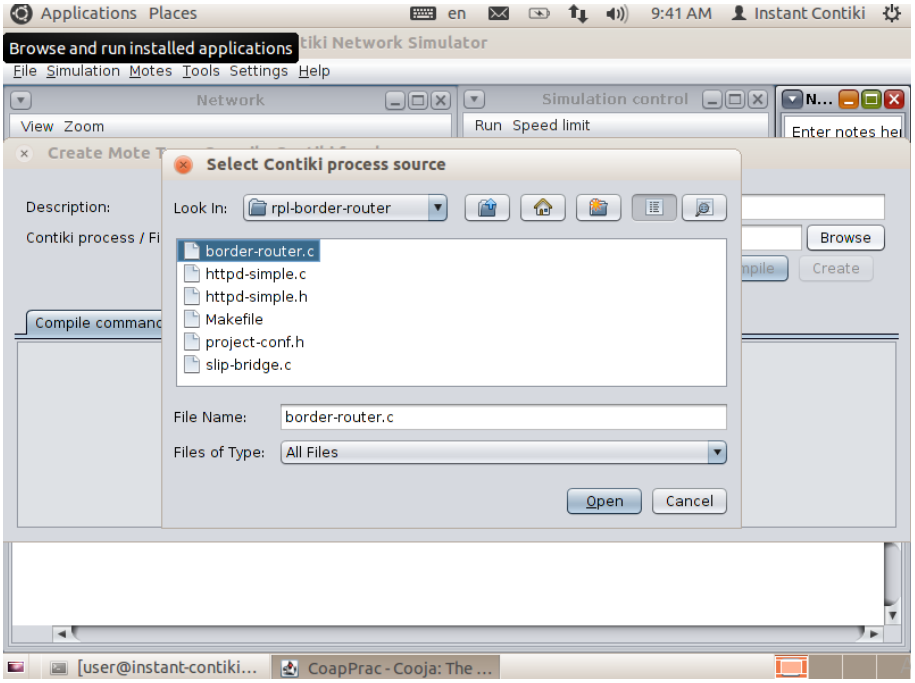


Figure 1.7 Adding Sky Mote

Now for selecting the process source in contiki browse the file rpl-border-router.c which is situated at the location:

/home/user/contiki-2.7/examples/ipv6/rpl-border-router/rpl-border-router.c

Then click on open:



Look In: rpl-border-router

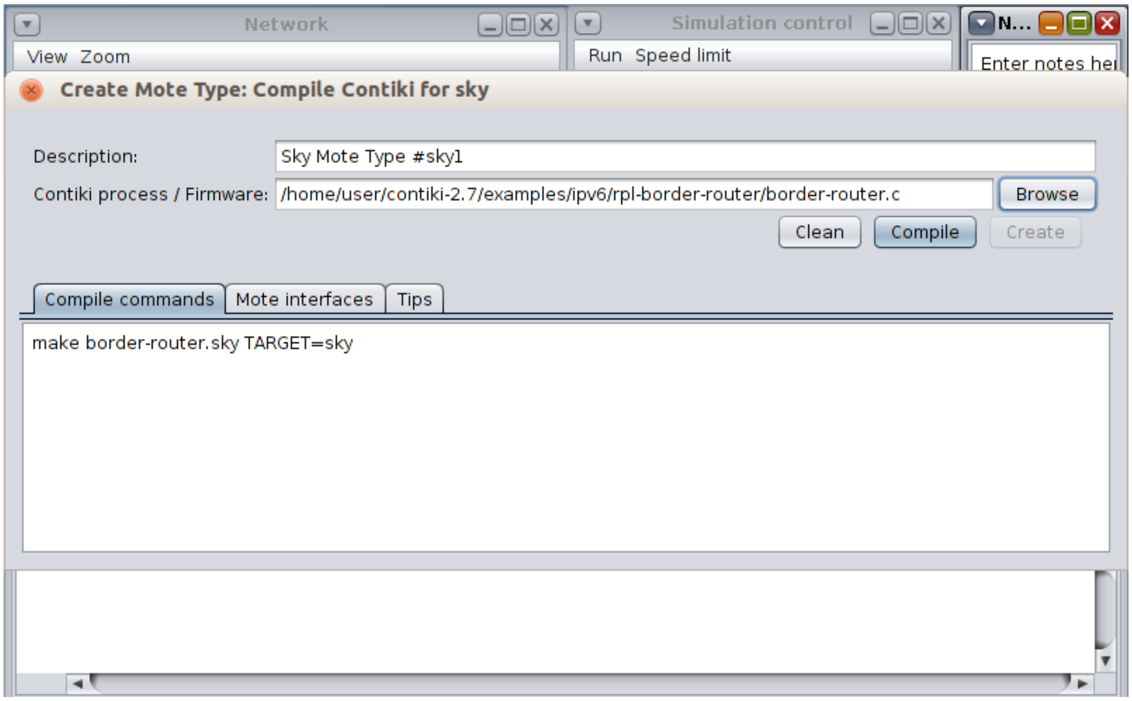
- border-router.c
- httpd-simple.c
- httpd-simple.h
- Makefile
- project-conf.h
- slip-bridge.c

File Name: border-router.c

Files of Type: All Files

Figure 1.8 Selecting border-router.c

Now Compile the Mote and then Click on Create:

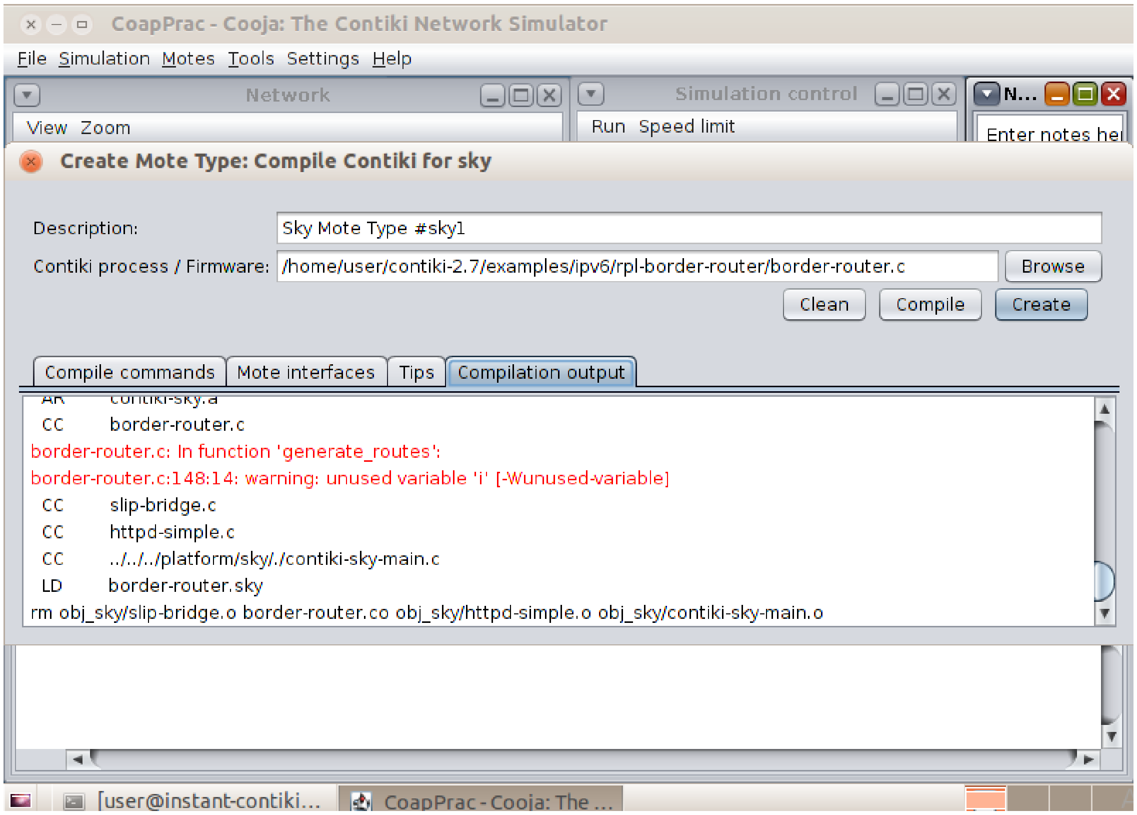


Description: Sky Mote Type #sky1

Contiki process / Firmware: /home/user/contiki-2.7/examples/ipv6/rpl-border-router/border-router.c

```
make border-router.sky TARGET=sky
```

Figure 1.9 Compile the mote



Description: Sky Mote Type #sky1

Contiki process / Firmware: /home/user/contiki-2.7/examples/ipv6/rpl-border-router/border-router.c

```
  AR      contiki-sky.a
  CC      border-router.c
border-router.c: In function 'generate_routes':
border-router.c:148:14: warning: unused variable 'i' [-Wunused-variable]
  CC      slip-bridge.c
  CC      httpd-simple.c
  CC      ../../../platform/sky/./contiki-sky-main.c
  LD      border-router.sky
rm obj_sky/slip-bridge.o border-router.co obj_sky/httpd-simple.o obj_sky/contiki-sky-main.o
```

Figure 1.10 Creating the mote

After Creating the mote then Click on the different Views in the Network tab of the Simulation:

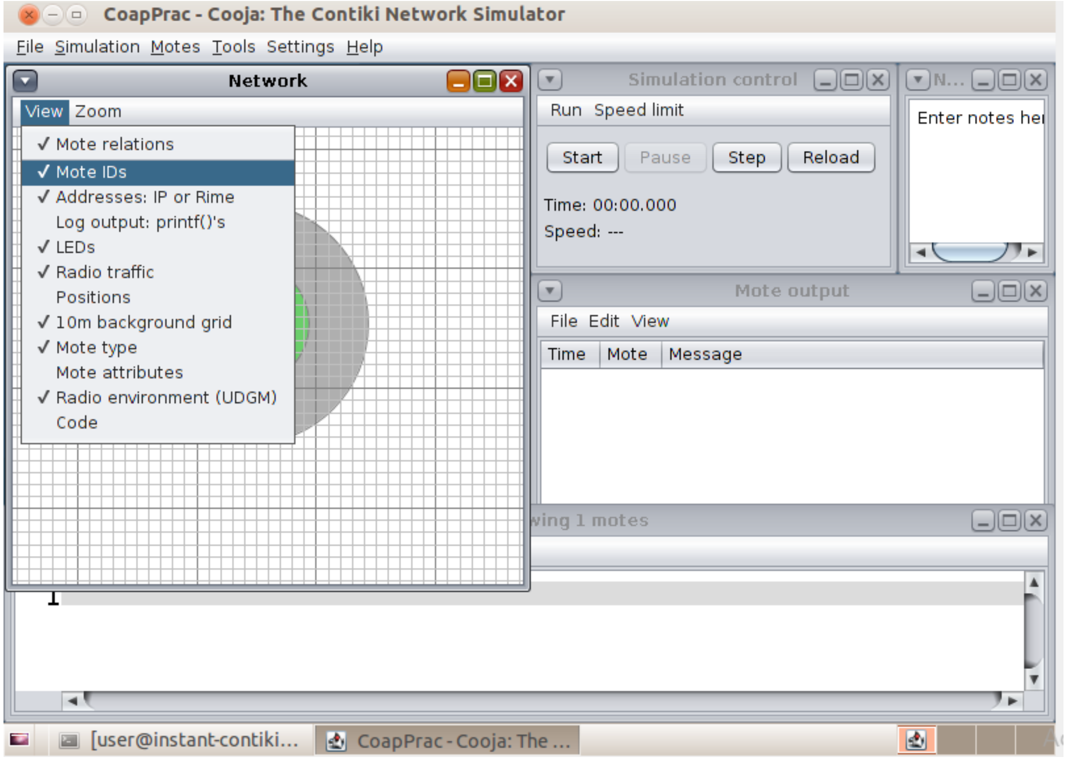


Figure 1.11 Selecting Views

Create Another sky mote and in this mote select the file as er-example-server.c file :

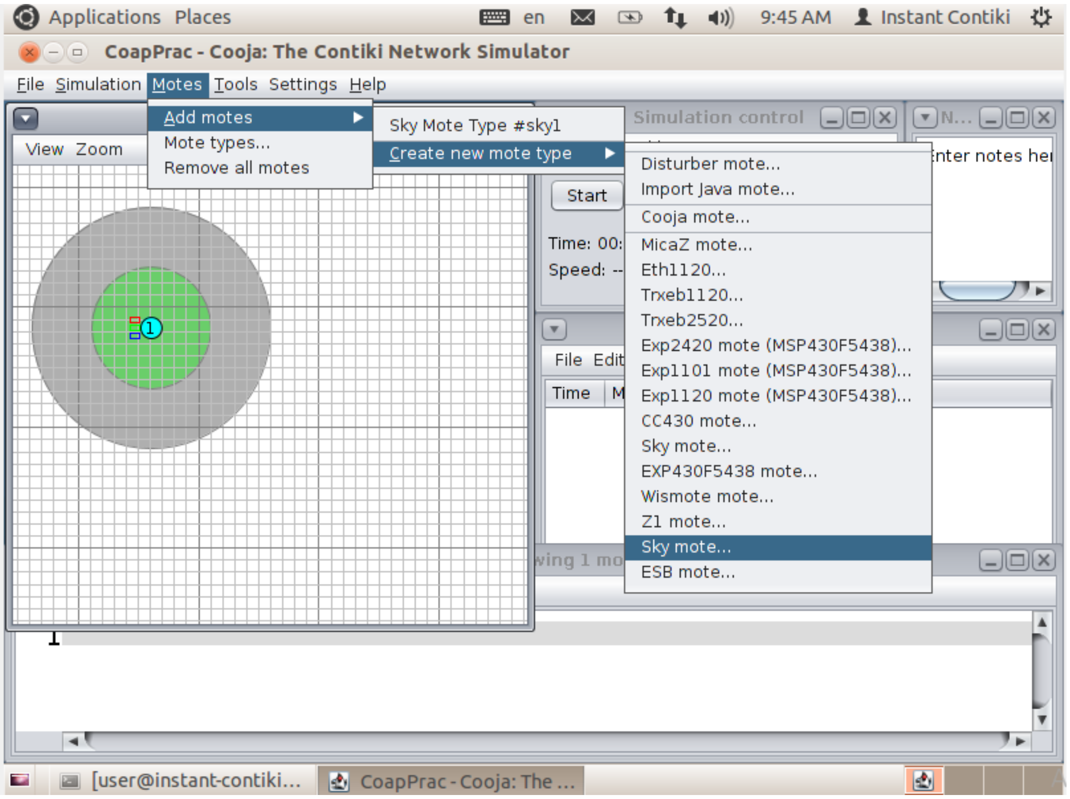
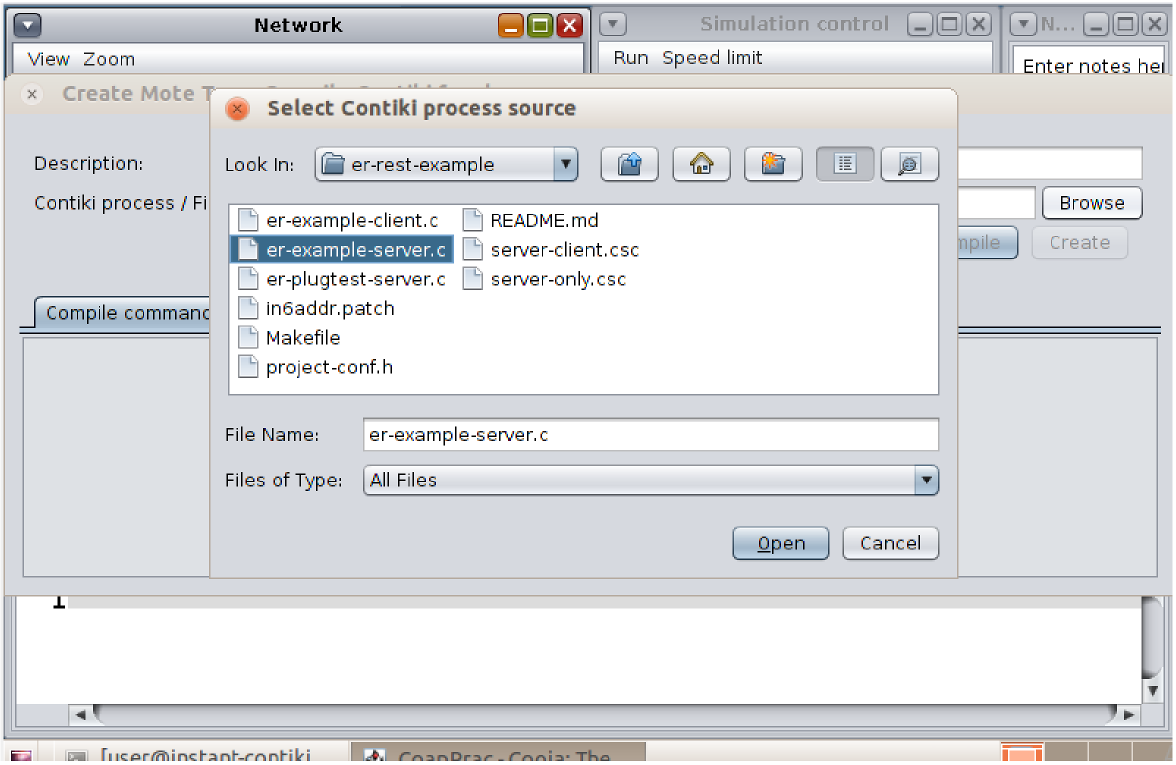


Figure 1.12 Create another mote

The Location of the file is :
/home/user/contiki-2.7/examples/er-rest-example/er-example-server.c:



Select Contiki process source

Look In: er-rest-example

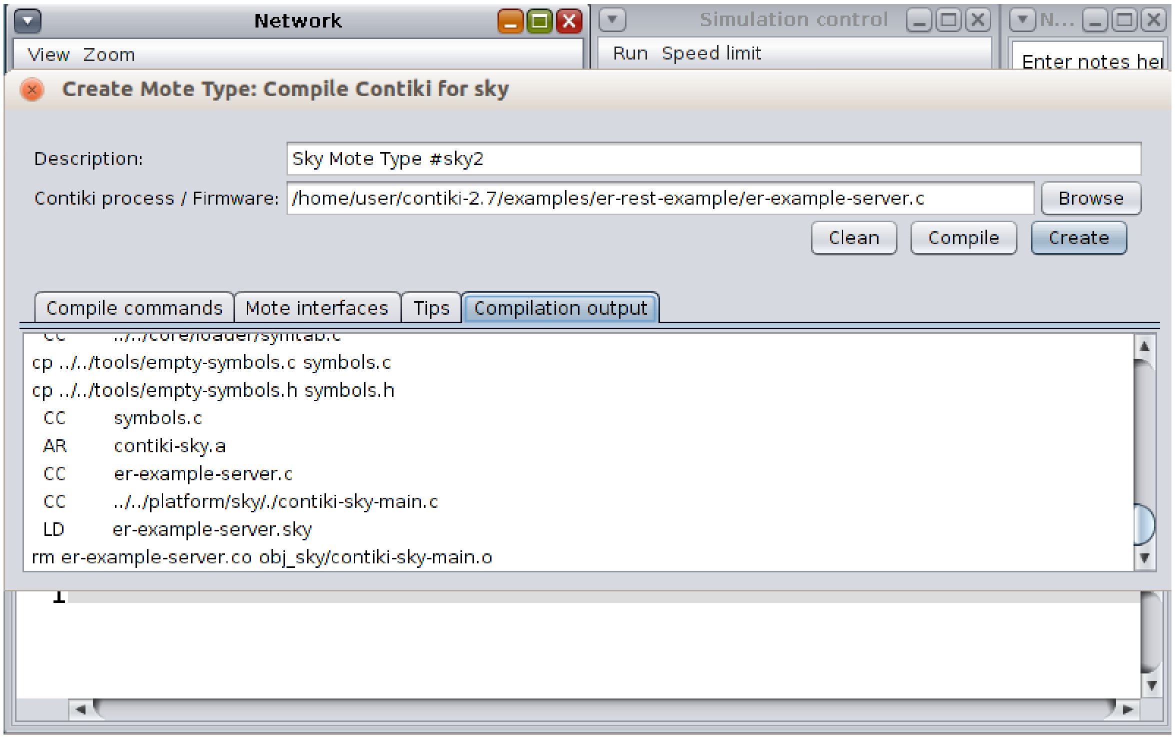
```
er-example-client.c      README.md
er-example-server.c      server-client.csc
er-plugtest-server.c     server-only.csc
in6addr.patch
Makefile
project-conf.h
```

File Name: er-example-server.c

Files of Type: All Files

Figure 1.13 Selecting er-example-server.c

Now Click on Compile and then Create:



Create Mote Type: Compile Contiki for sky

Description: Sky Mote Type #sky2

Contiki process / Firmware: /home/user/contiki-2.7/examples/er-rest-example/er-example-server.c

```
  CC      ../../core/loader/symtab.c
cp ../../tools/empty-symbols.c symbols.c
cp ../../tools/empty-symbols.h symbols.h
  CC      symbols.c
  AR      contiki-sky.a
  CC      er-example-server.c
  CC      ../../platform/sky/./contiki-sky-main.c
  LD      er-example-server.sky
rm er-example-server.co obj_sky/contiki-sky-main.o
```

Figure 1.14 Compiling and creating the mote

When we click on the 1st mote the area covered is seen and then we need to make the socket server:

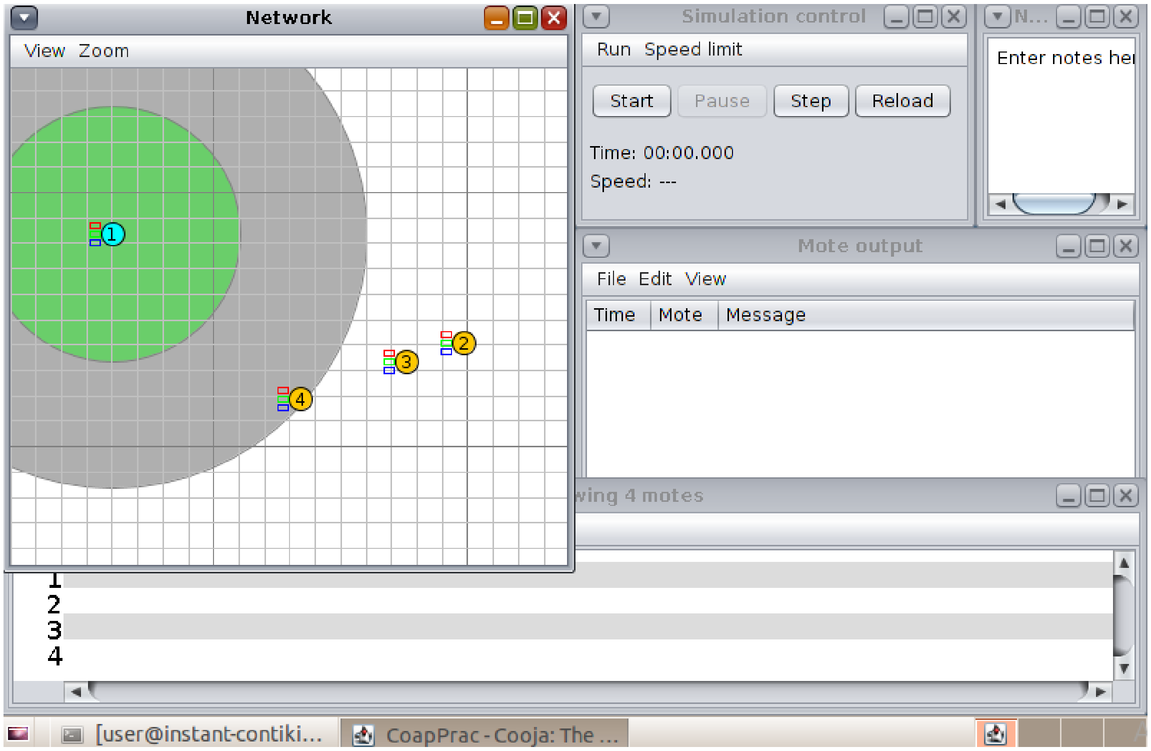
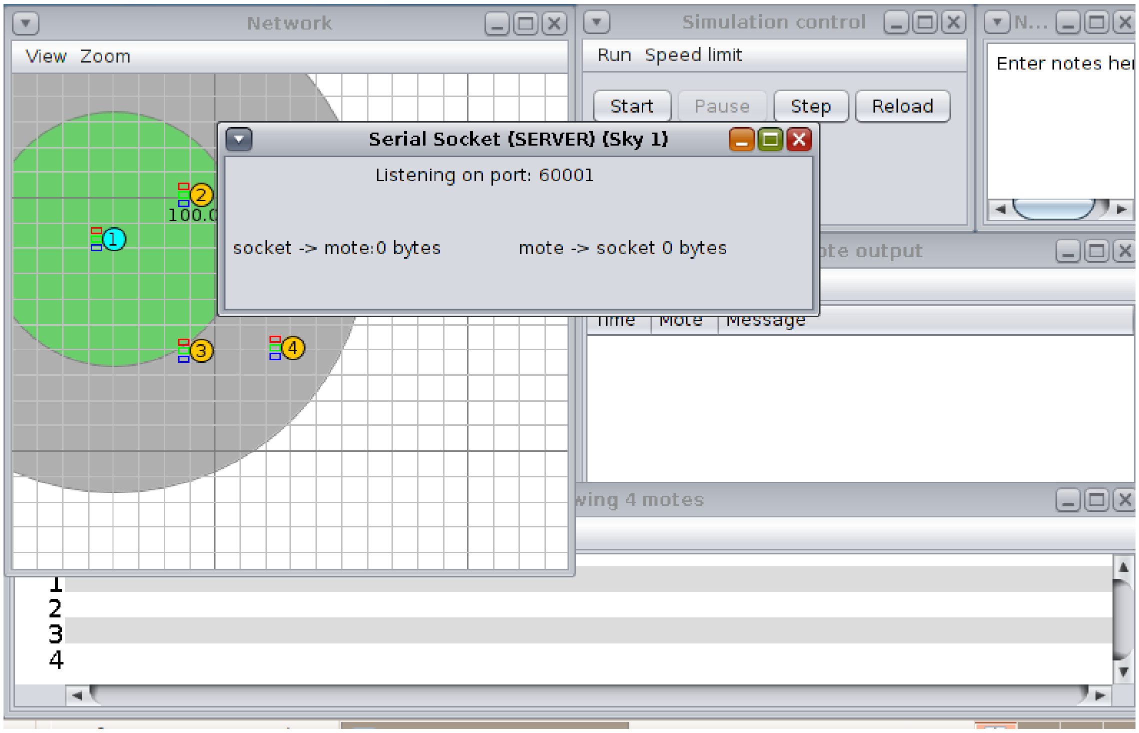


Figure 1.15 Network View

Right Click on the 1st mote and then Click on Mote tools for Sky 1 and then click on Serial Socket(SERVER) and thus the below result appears:

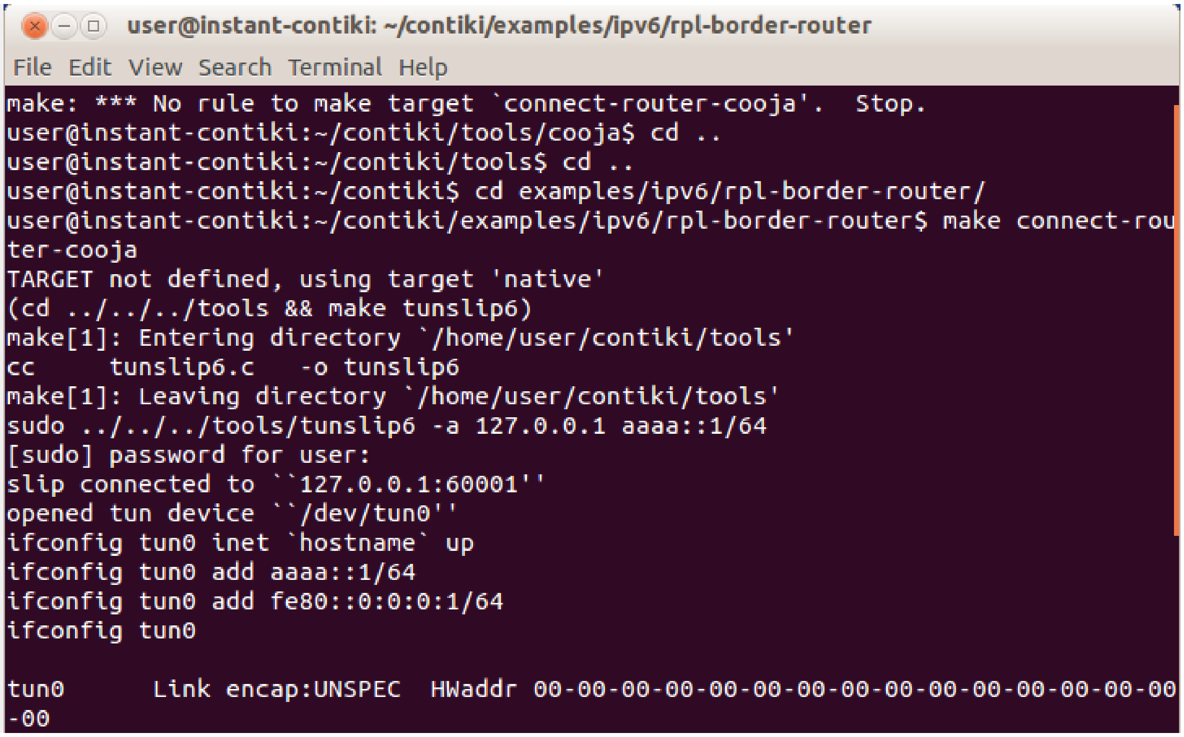


Listening on port: 60001

socket -> mote:0 bytes mote -> socket 0 bytes

Figure 1.16 Serial Socket(SERVER)

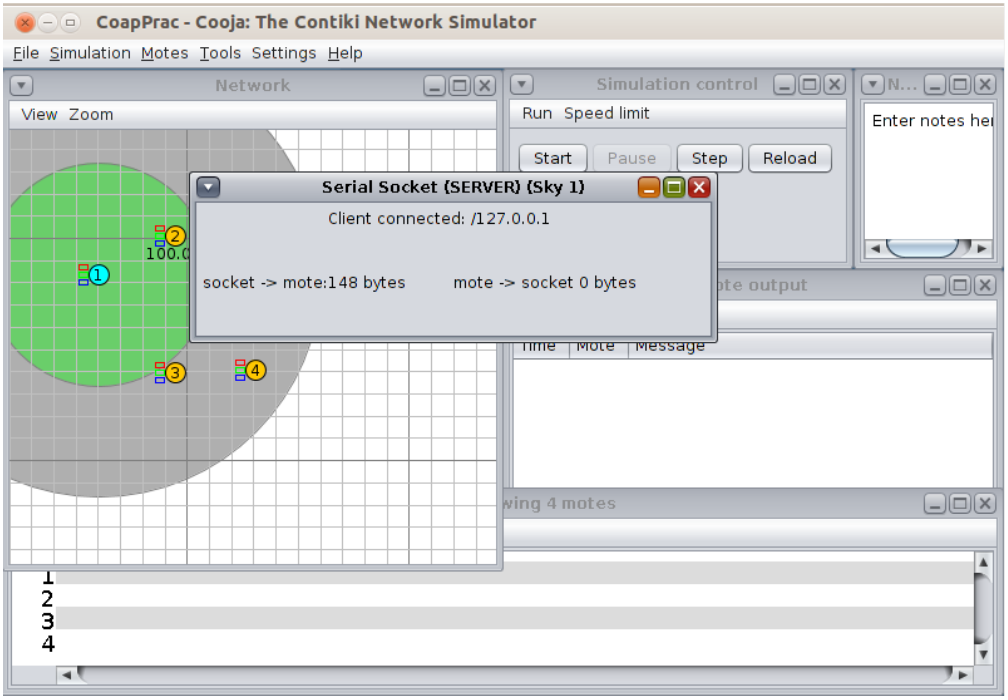
Now for connecting the router we need to write the command as make connect-router-cooja in the rpl-border-router folder in the terminal :


user@instant-contiki: ~/contiki/examples/ipv6/rpl-border-router
File Edit View Search Terminal Help

```
make: *** No rule to make target `connect-router-cooja'.  Stop.
user@instant-contiki:~/contiki/tools/cooja$ cd ..
user@instant-contiki:~/contiki/tools$ cd ..
user@instant-contiki:~/contiki$ cd examples/ipv6/rpl-border-router/
user@instant-contiki:~/contiki/examples/ipv6/rpl-border-router$ make connect-rou
ter-cooja
TARGET not defined, using target 'native'
(cd ../../../tools && make tunslip6)
make[1]: Entering directory `/home/user/contiki/tools'
cc     tunslip6.c   -o tunslip6
make[1]: Leaving directory `/home/user/contiki/tools'
sudo ../../../tools/tunslip6 -a 127.0.0.1 aaaa::1/64
[sudo] password for user:
slip connected to ``127.0.0.1:60001''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` up
ifconfig tun0 add aaaa::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
```

Figure 1.17 IP address of the server

The Client connected are seen in the Serial Socket (SERVER) in the simulation screen:

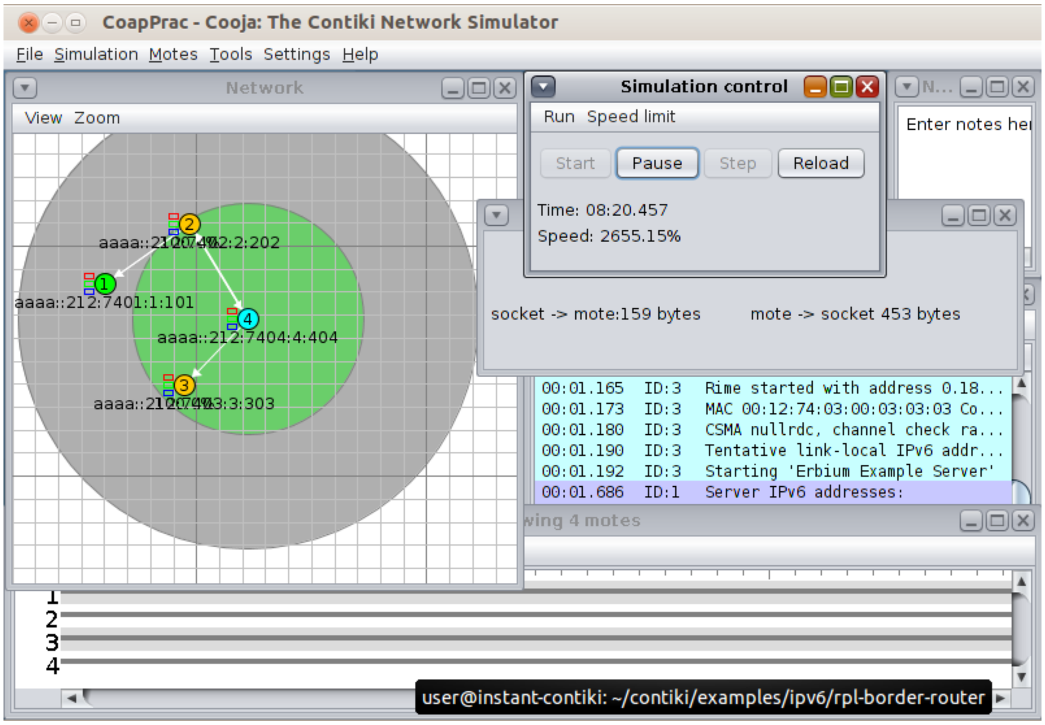


Client connected: /127.0.0.1

socket -> mote:148 bytes   mote -> socket 0 bytes

Figure 1.18 Sky 1

Now start the Simulation and thus the connection and message transfer is seen in the Network Tab of the Simulation Screen:



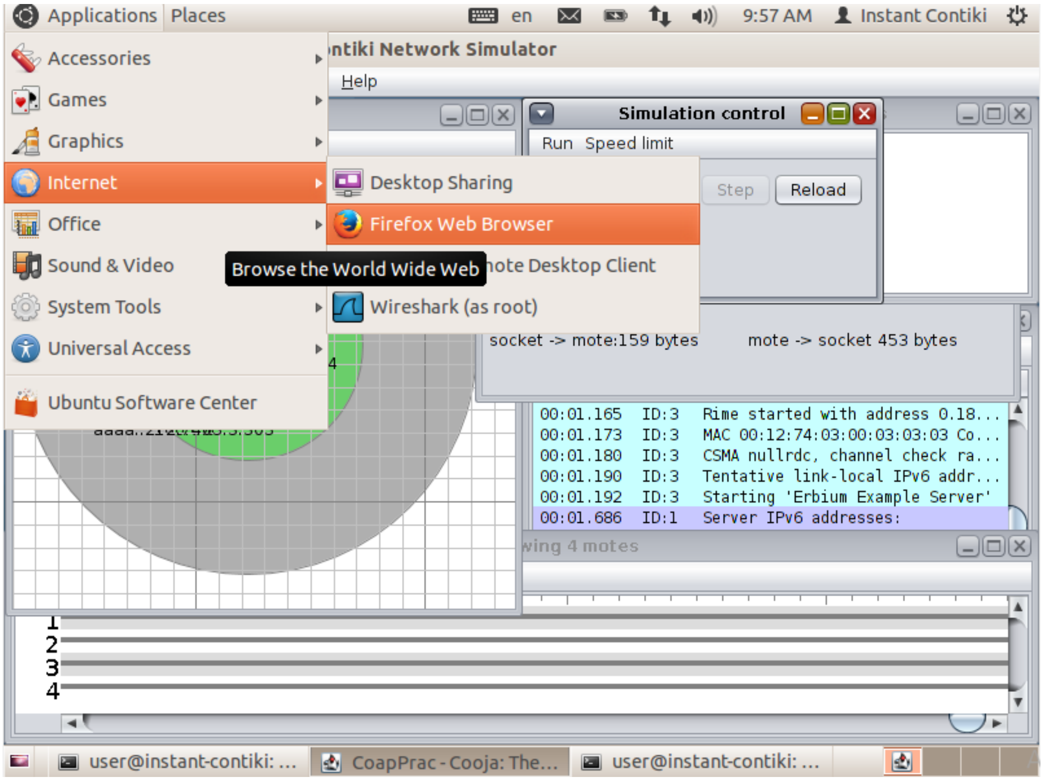
Time: 08:20.457
Speed: 2655.15%

socket -> mote:159 bytes mote -> socket 453 bytes

```
00:01.165  ID:3  Rime started with address 0.18...
00:01.173  ID:3  MAC 00:12:74:03:00:03:03:03 Co...
00:01.180  ID:3  CSMA nullrdc, channel check ra...
00:01.190  ID:3  Tentative link-local IPv6 addr...
00:01.192  ID:3  Starting 'Erbium Example Server'
00:01.686  ID:1  Server IPv6 addresses:
```

Figure 1.19 Simulation Started

Click on the Applications and there click on Internet and click Firefox Web browser:

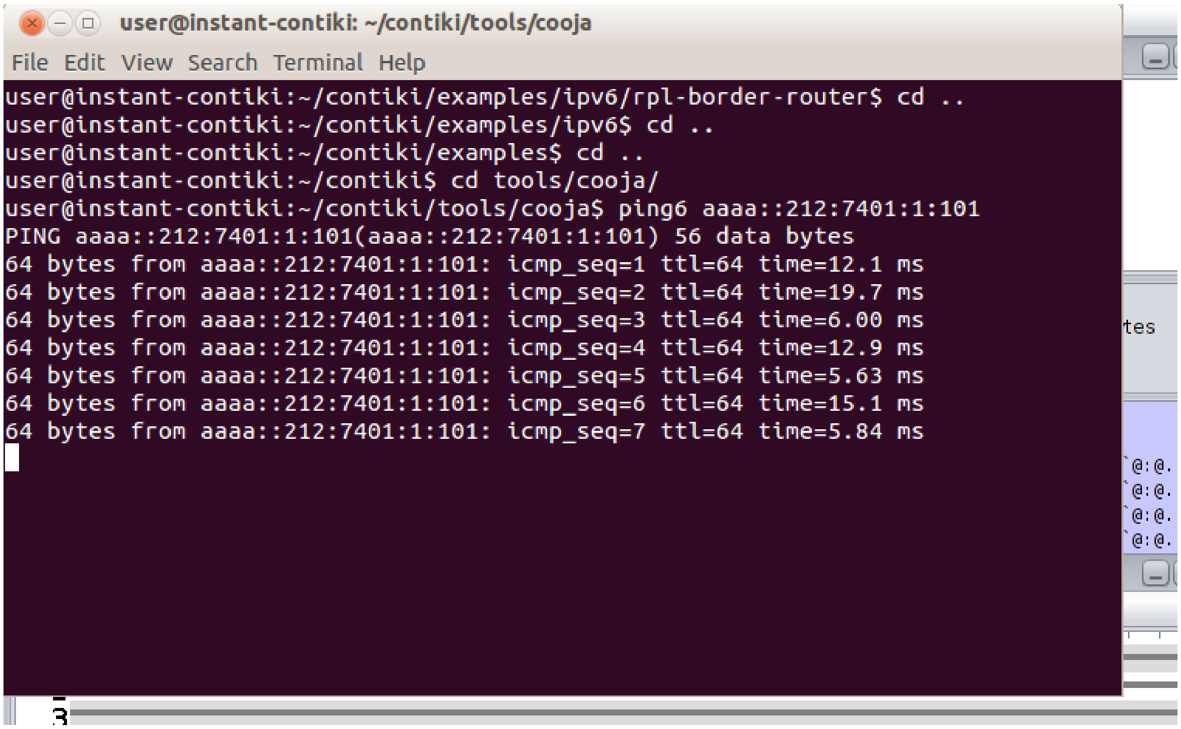


socket -> mote:159 bytes mote -> socket 453 bytes

```
00:01.165  ID:3  Rime started with address 0.18...
00:01.173  ID:3  MAC 00:12:74:03:00:03:03:03 Co...
00:01.180  ID:3  CSMA nullrdc, channel check ra...
00:01.190  ID:3  Tentative link-local IPv6 addr...
00:01.192  ID:3  Starting 'Erbium Example Server'
00:01.686  ID:1  Server IPv6 addresses:
```

Figure 1.20 Open Firefox

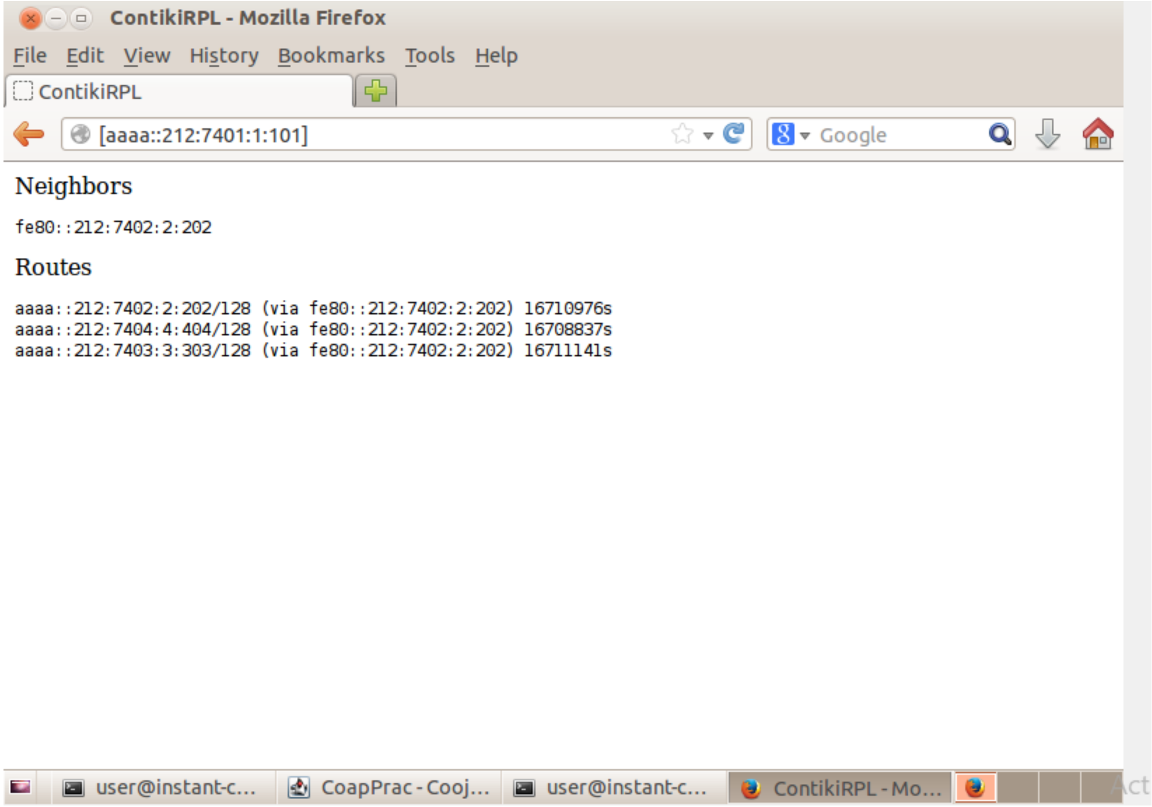
Run the ping6 command with the ip address of the mote that is selected as the Socket server:


user@instant-contiki: ~/contiki/tools/cooja
File Edit View Search Terminal Help

```
user@instant-contiki:~/contiki/examples/ipv6/rpl-border-router$ cd ..
user@instant-contiki:~/contiki/examples/ipv6$ cd ..
user@instant-contiki:~/contiki/examples$ cd ..
user@instant-contiki:~/contiki$ cd tools/cooja/
user@instant-contiki:~/contiki/tools/cooja$ ping6 aaaa::212:7401:1:101
PING aaaa::212:7401:1:101(aaaa::212:7401:1:101) 56 data bytes
64 bytes from aaaa::212:7401:1:101: icmp_seq=1 ttl=64 time=12.1 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=2 ttl=64 time=19.7 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=3 ttl=64 time=6.00 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=4 ttl=64 time=12.9 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=5 ttl=64 time=5.63 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=6 ttl=64 time=15.1 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=7 ttl=64 time=5.84 ms
```

Figure 1.21 ping6 in terminal

Writing the ip in the web browser shows the Neighbours and the Routes in the page:



Neighbors

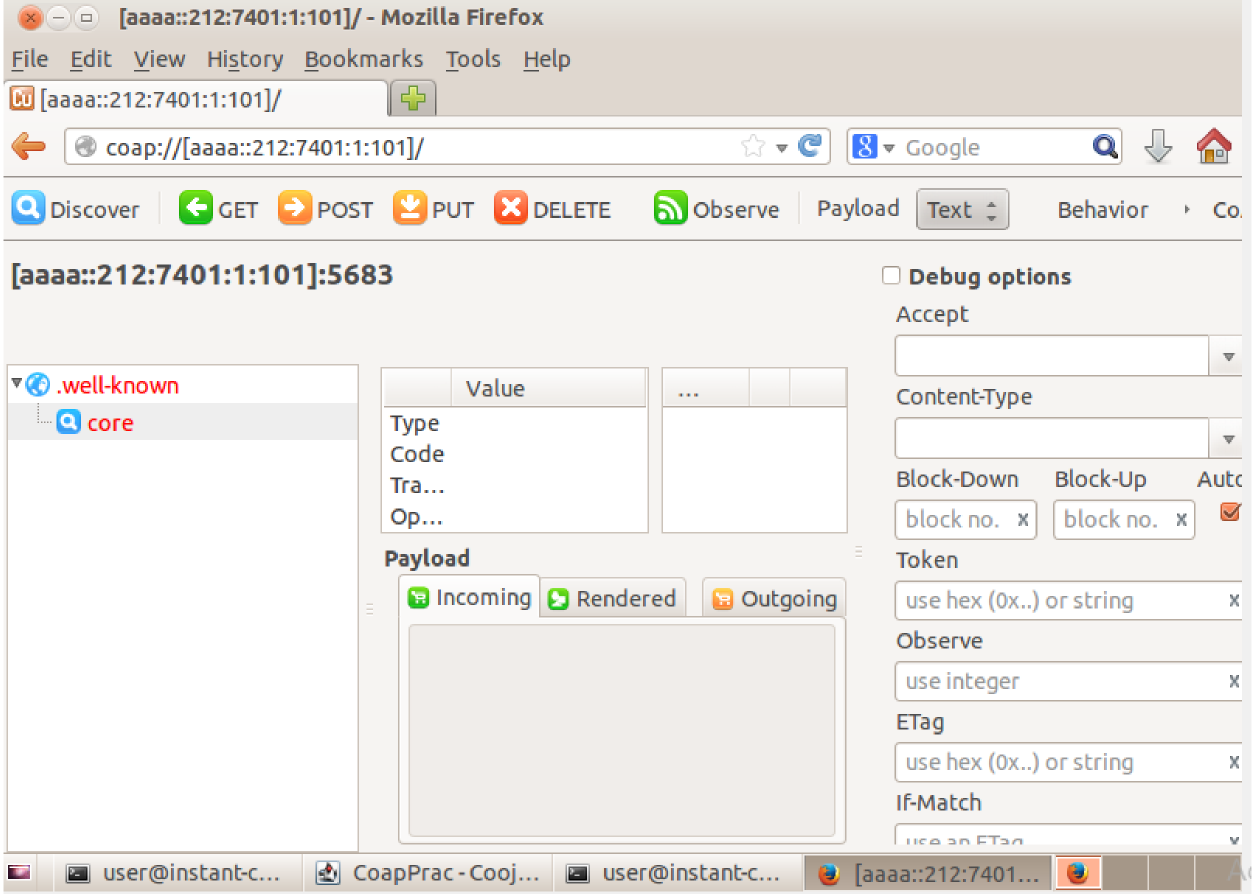
```
fe80::212:7402:2:202
```

Routes

```
aaaa::212:7402:2:202/128 (via fe80::212:7402:2:202) 16710976s
aaaa::212:7404:4:404/128 (via fe80::212:7402:2:202) 16708837s
aaaa::212:7403:3:303/128 (via fe80::212:7402:2:202) 16711141s
```

Figure 1.22 Neighbours and Routes

Now Writing the coap:// in the web browser it will result in the below web browser and from here we can get and post :



coap://[aaaa::212:7401:1:101]/

[aaaa::212:7401:1:101]:5683

Figure 1.23 CoAp in Firefox

Conclusion: We learned, understood and implemented the CoAp in the Contiki and thus learned how the CoAp works and in the firefox web browser and the how the server works.