PRACTICAL-4

AIM

Create a scenario by adding some motes. Do the simulation for the same. Also observe the result for said scenario. Scenario: When one mote sends the signal then led should turn green while one receives then it should show red color. Simulate BGP and RPL protocol in Cooja.

THEORY

BGP

- Border Gateway Protocol (BGP) is a standardized exterior gateway protocol designed to exchange routing and reachability information between autonomous systems (AS) on the Internet.
- The protocol is often classified as a path vector protocol but is sometimes also classed as a distance-vector routing protocol."
- BGP (a.k.a. Border Gateway Protocol) is the routing method that enables the Internet to function. Without it, we wouldn't be able to do a Google search or send an email.
- Each BGP speaker, which is called a "peer", exchanges routing information with its neighboring peers in the form of network prefix announcements. This way, an AS doesn't need to be connected to another AS to know its network prefix.
- The BGP decision-making mechanism analyzes all the data and sets one of its peers as the next stop, to forward packets for a certain destination.
- Each peer manages a table with all the routes it knows for each network and propagates that information to its neighboring autonomous systems.
- In this way, BGP allows an AS to collect all the routing information from its neighboring autonomous systems and "advertise" that information further. Each peer transfers the information internally inside its own autonomous system.
- Just like in real life, usually more than one route exists to reach a given destination. BGP is responsible for determining the most suitable route according to the information collected and an organization's routing policy, which is based on cost, reliability, speed, etc.

RPL

- RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. It is a proactive protocol based on distance vectors and operates on IEEE 802.15.4, optimized for multi-hop and many-to-one communication, but also supports one-to-one messages.

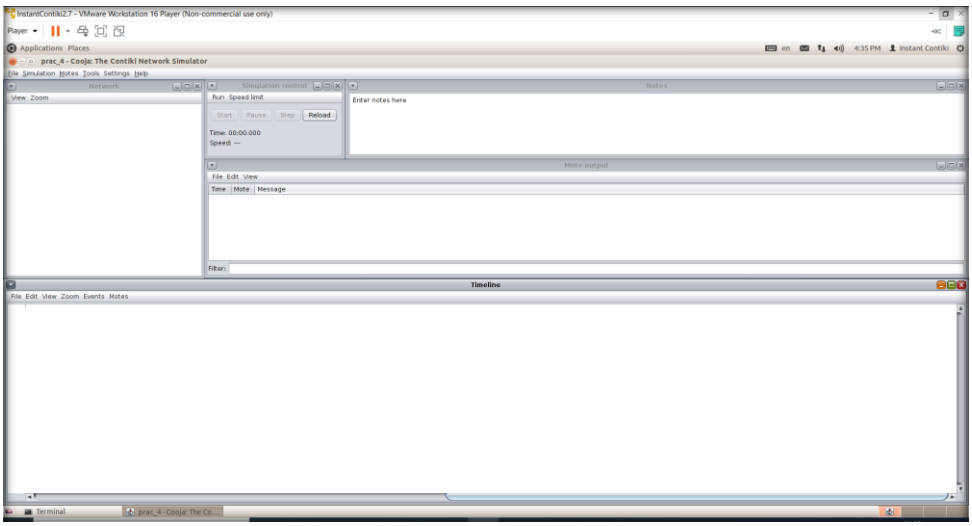
- This protocol is specified in RFC 6550 with special applications in RFCs 5867, 5826, 5673 and 5548. RPL can support a wide variety of link layers, including those with limitations, with potential losses or that are used in devices with limited resources.

- This protocol can quickly create network routes, share routing knowledge and adapt the topology in an efficient way.

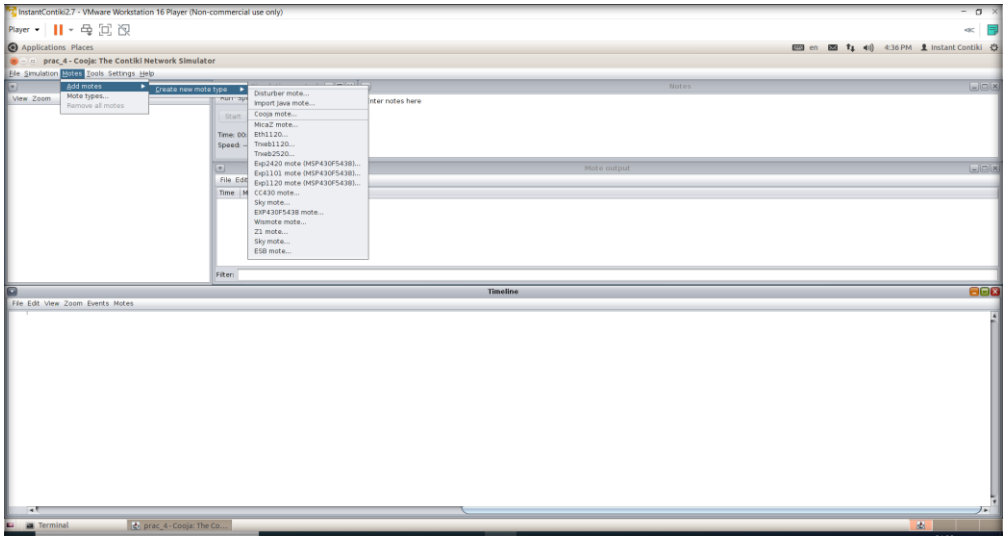
- The implementation of the RPL protocol occurs in wireless sensors and networks, the most used operating system for its implementation is Contiki which is a small open source operating system developed for use in a number of small systems ranging from 8-bit computers to integrated systems on microcontrollers, including sensor network nodes.

PRACTICAL IMPLEMENTATION

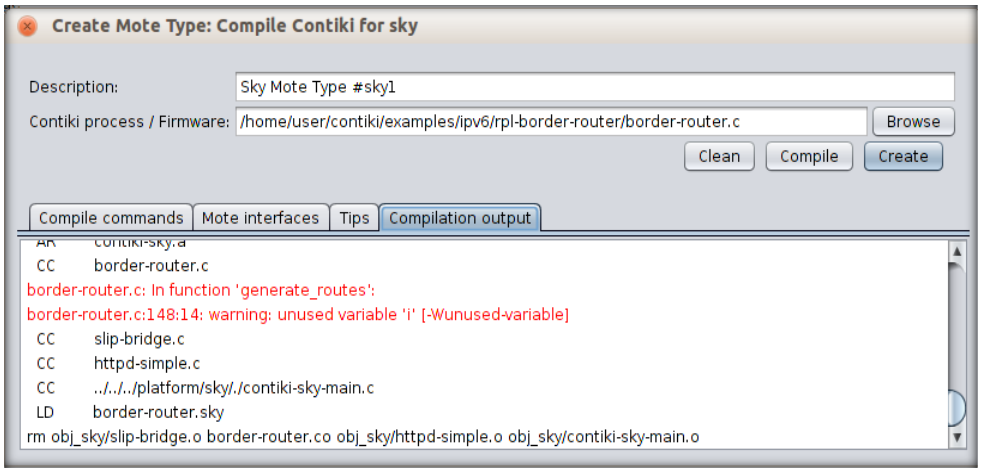
- First of all, we will open Cooja simulator and create new simulator.



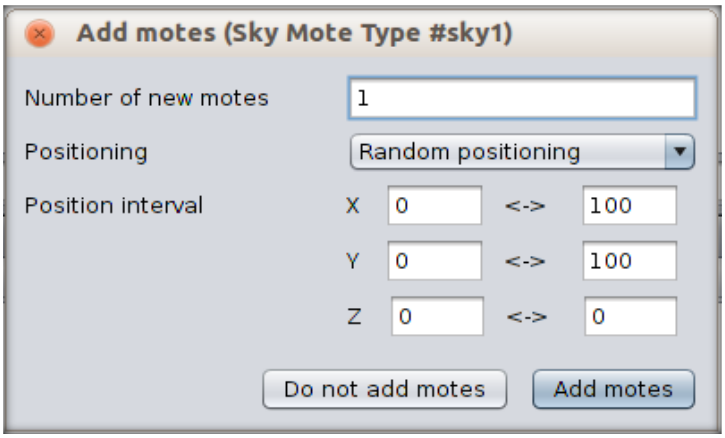
- Then, we will first add the router mote, so we will go to Motes > Create new mote type > Sky mote.



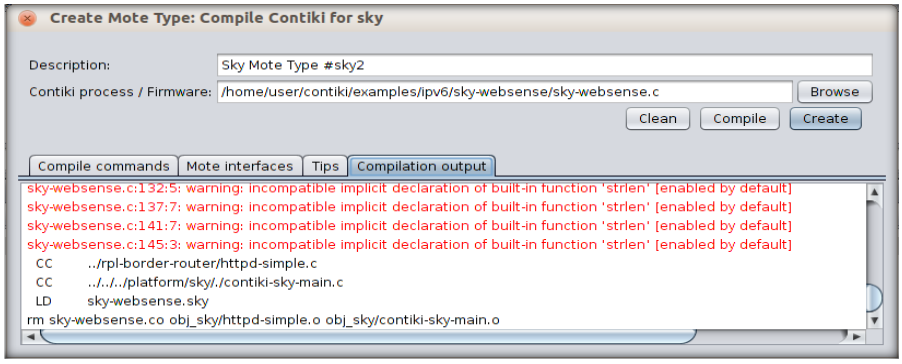
- We will browse through example folder to find ipv6 > rpl-borer-router > border-router.c.
- We will compile it and press create.



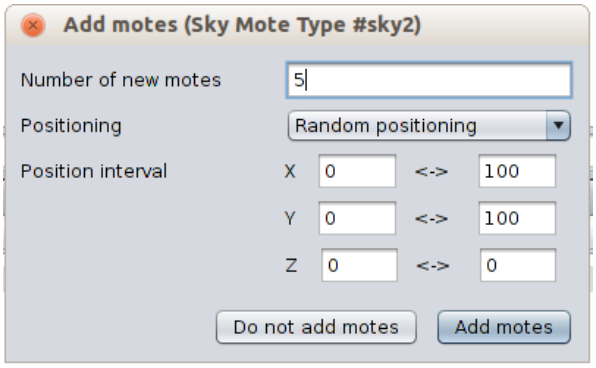
- We will need only 1 router so number of new motes will be 1 only.



- To create receiving motes, we will follow the same process to add sky motes.
- This time, we will browse to ipv6 > sky-websense > sky-websense.c.

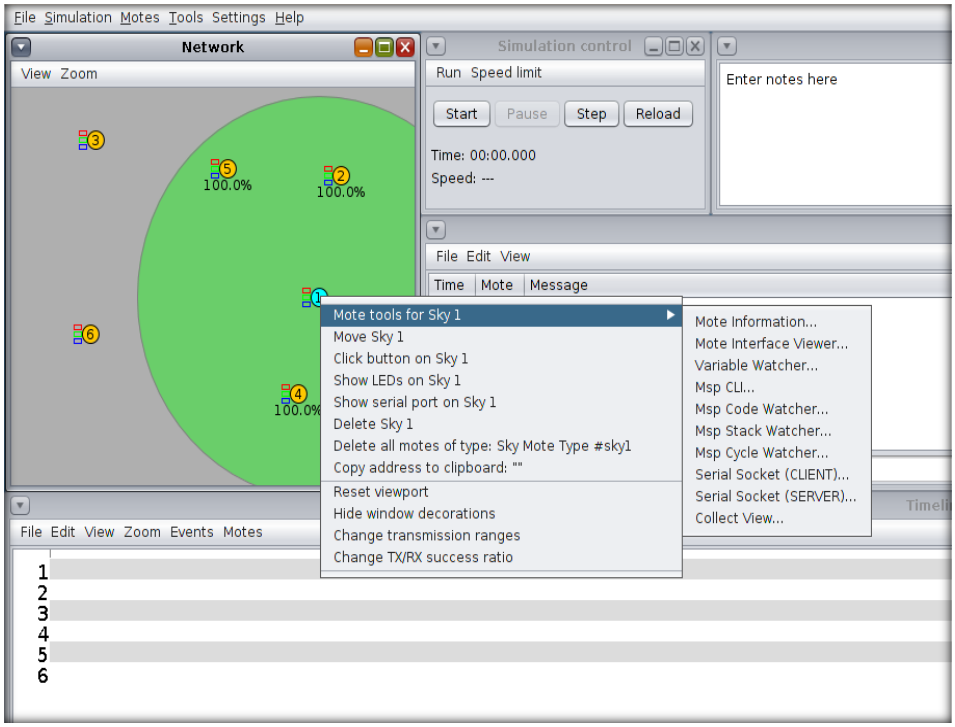


- We will need more receivers for better visualization so we will add 5 motes.

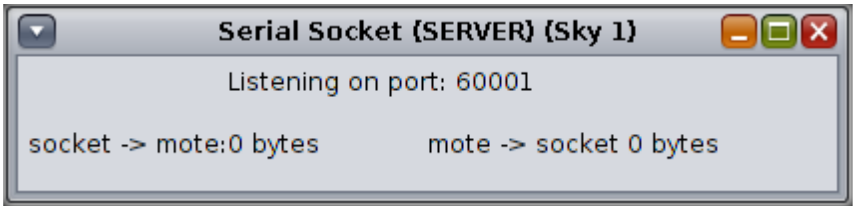


- When we will add motes, we will be able to see a screen like this.
- By clicking on the mote, you will be able to see its range.
- We can change some view options as our convenience.

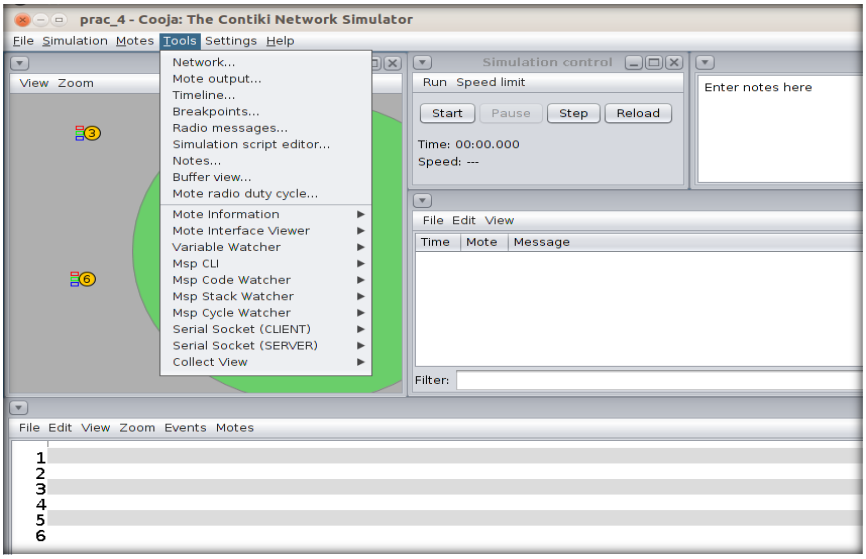
- To see the traffic, we will right click on server and go to Mote tools for Sky 1 > Serial Socket (SERVER)



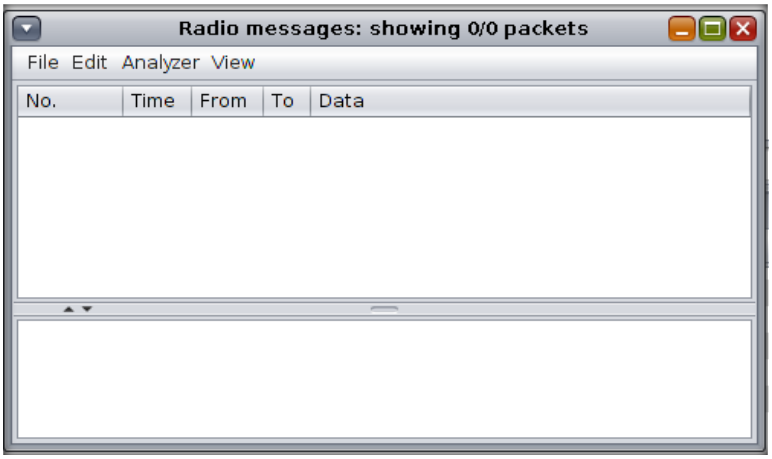
- We will be able to see a new window like below.



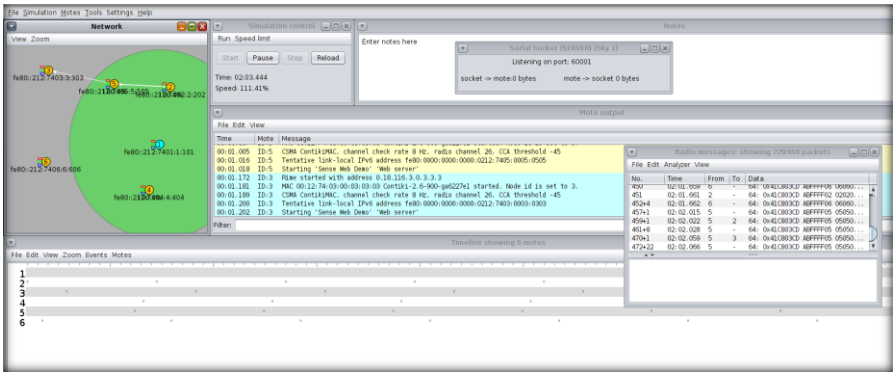
- To get more information about traffic, we can go to tools > Radio messages.



- We will be able to see another window like below.



- Then, we will finally start the simulator to see the radio traffic.



- Now, to visualize the content of simulator in browser, we will run the following command into different terminal.
- Commands:

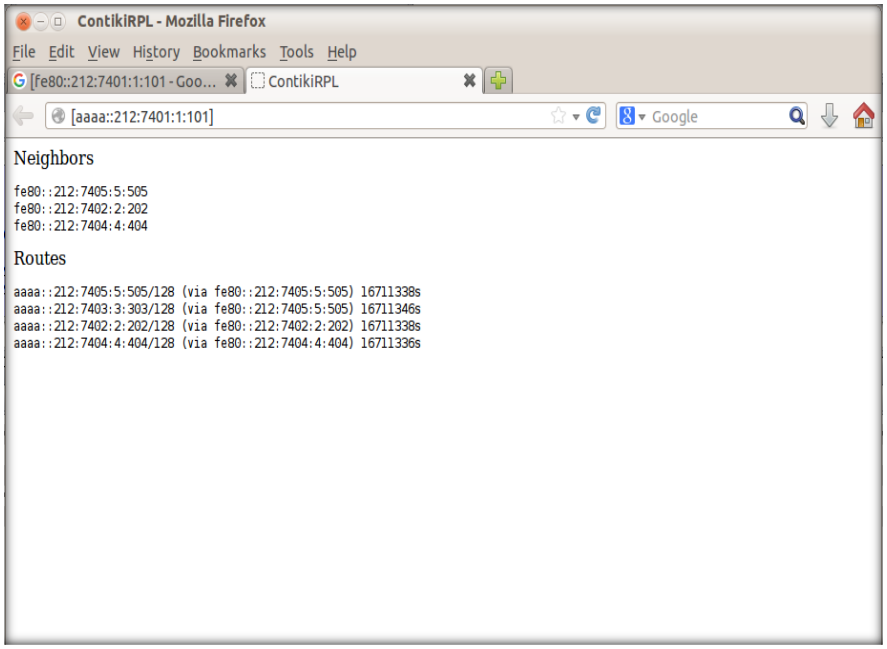
cd Contiki/examples/ipv6/rpl-border-router/
Make connect-router-cooja

```
user@instant-contiki:~/contiki/examples/ipv6/rpl-border-router$ make connect-router-cooja
TARGET not defined, using target 'native'
sudo ../../../tools/tunslip6 -a 127.0.0.1 aaaa::1/64
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
          inet addr:127.0.1.1  P-t-P:127.0.1.1  Mask:255.255.255.255
          inet6 addr: fe80::1/64 Scope:Link
          inet6 addr: aaaa::1/64 Scope:Global
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

*** Address:aaaa::1 => aaaa:0000:0000:0000
Got configuration message of type P
Setting prefix aaaa::
Server IPv6 addresses:
 aaaa::212:7401:1:101
 fe80::212:7401:1:101
```

- Now, if we write ip address of router in browser like firefox, we will be able to see its neighbors information.



CONCLUSION

In this practical, I learnt regarding how to implement BGP and RPL protocol in Cooja.