

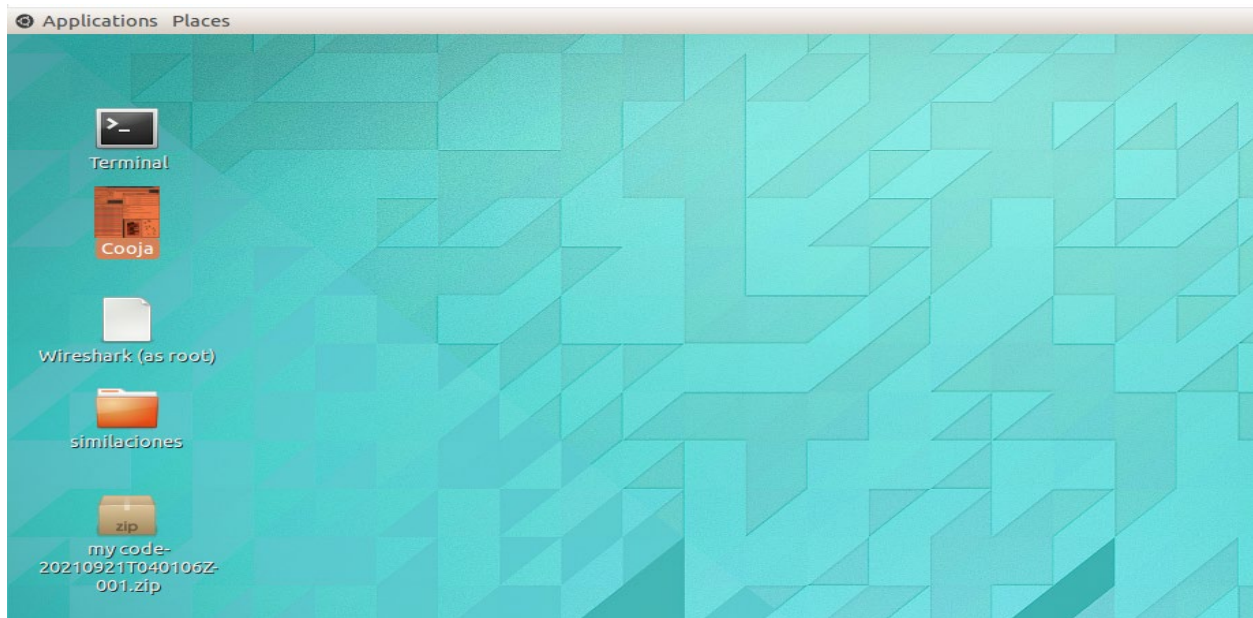
Apéndice E Procedimiento para experimentación en Cooja.

Para realizar una simulación y monitorear las métricas de esta se usa el aplicativo del sistema operativo *Collect View*, como firmware se usará a la aplicación *rpl collect* que está diseñada para recolectar las métricas de la red simulada y pasarlas al aplicativo *Collect View*, dicha simulación cuenta con dos tipos de nodos, nodo raíz y nodo remitente, los nodos remitentes envían mensajes al nodo receptor que es el nodo raíz.

A continuación, se listará los pasos de cómo se implementa una simulación.

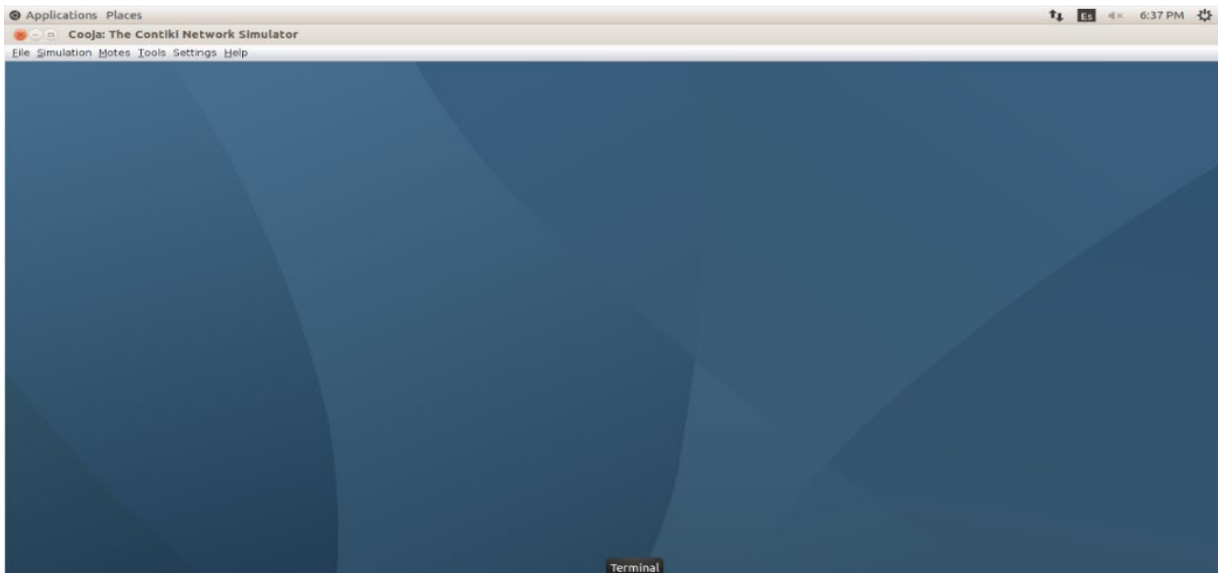
1. En la interfaz gráfica del sistema operativo Contiki se procede a abrir el simulador Cooja dando doble clic en el icono ubicado en el escritorio.

Figura 1 Paso 1 de implementación de simulación



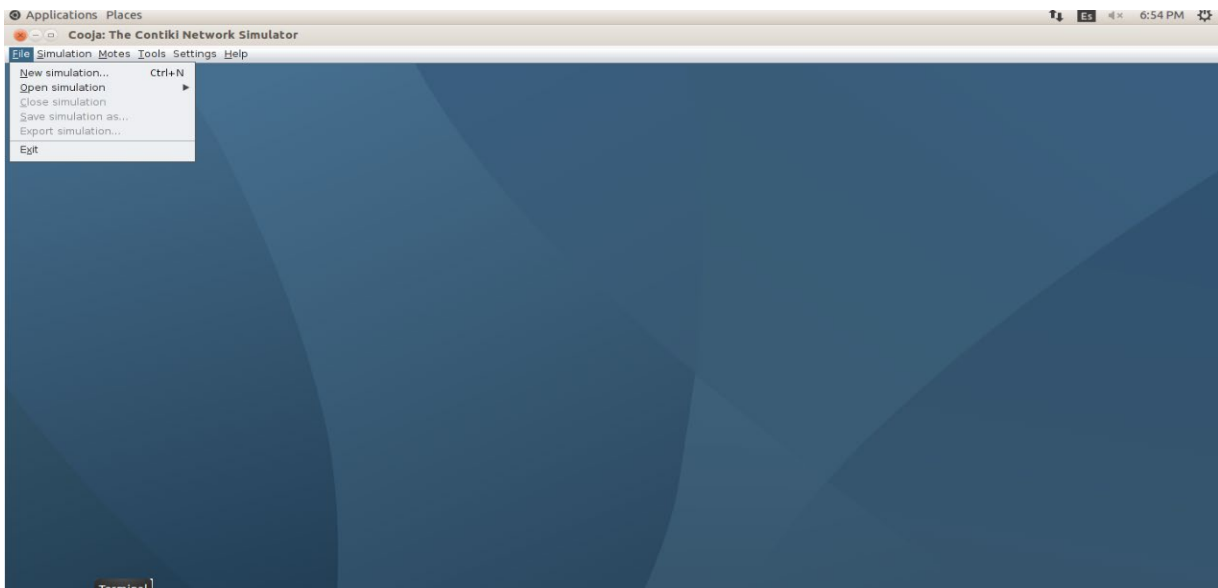
2. Se abre la siguiente interfaz gráfica con la opciones *File*, *Simulation*, *Motes*, *Tools*, *Settings* y *Help*.

Figura 2 Paso 2 de implementación de simulación



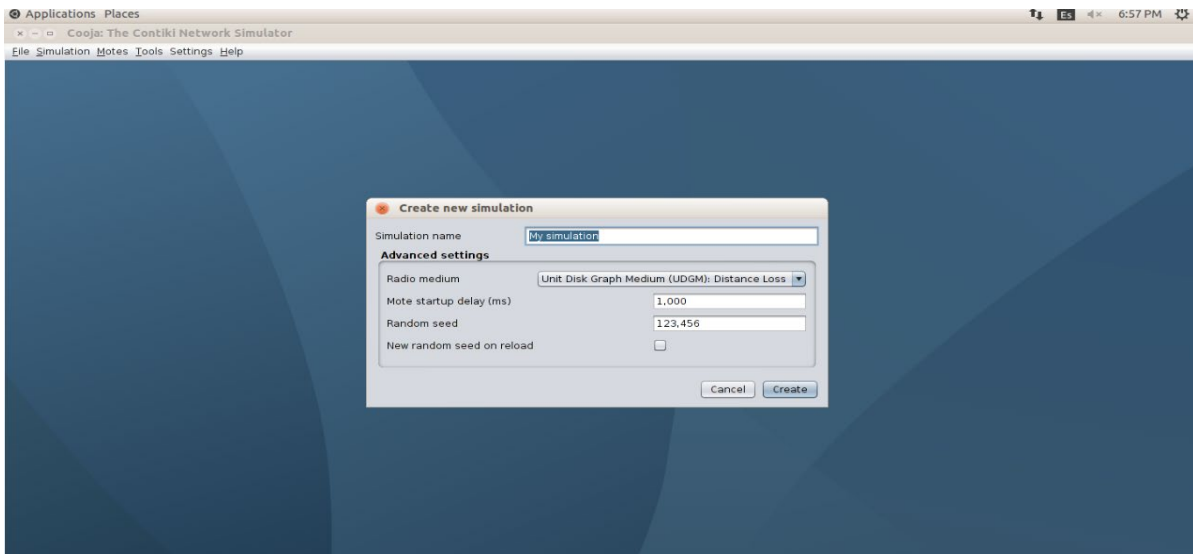
3. Se da clic en “file” y seguido de esto clic en “New simulation”.

Figura 3 Paso 3 de implementación de simulación



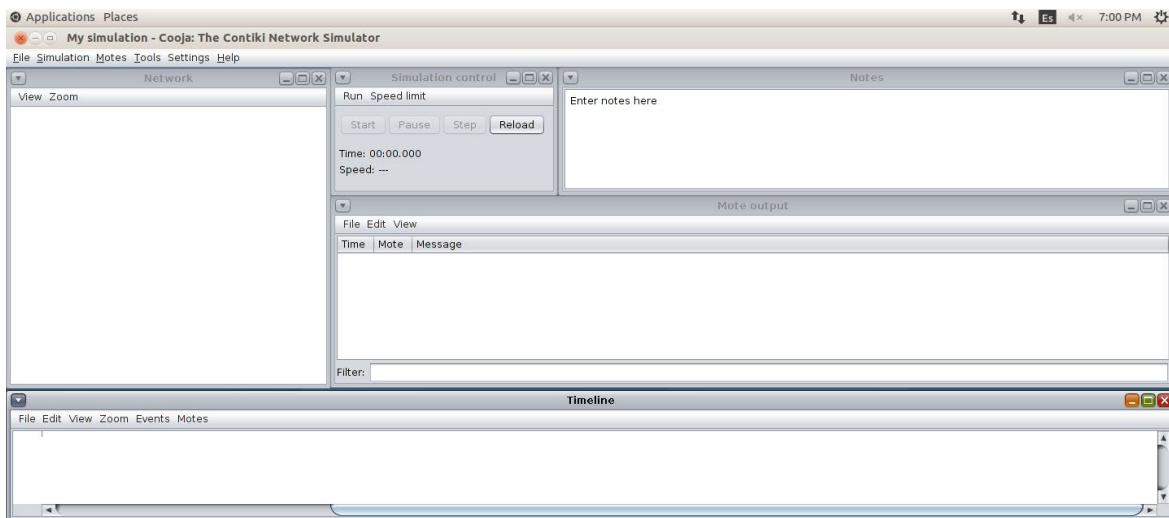
4. Luego se despliega un recuadro donde se asigna el nombre que se desea dar a la simulación y posteriormente se da clic en “create”.

Figura 4 Paso 4 de implementación de simulación



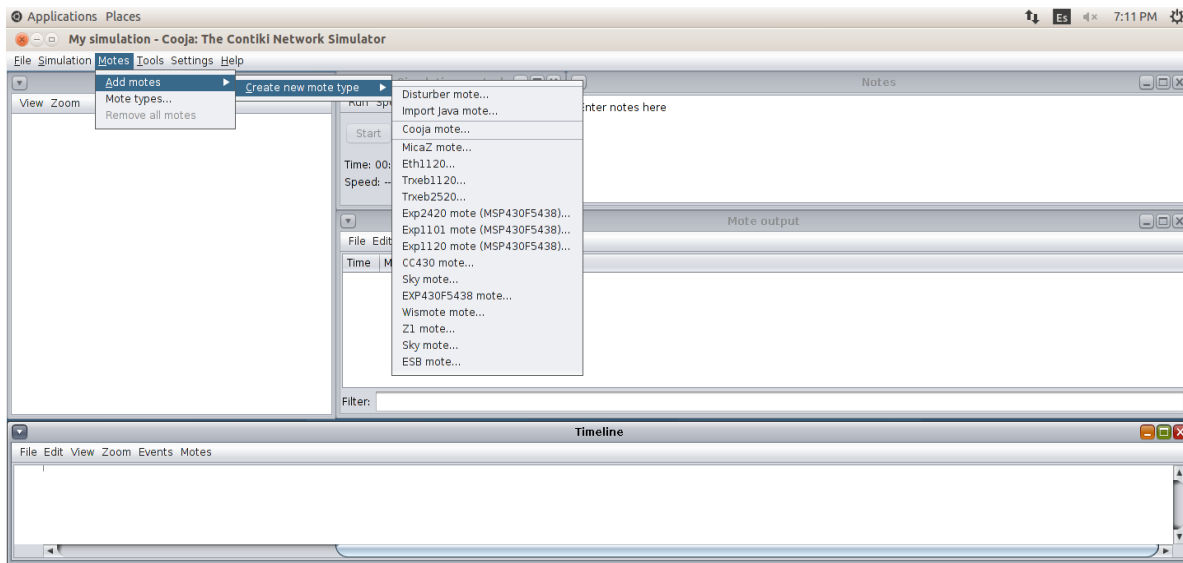
5. Se despliega la siguiente interfaz gráfica

Figura 5 Paso 5 de implementación de simulación



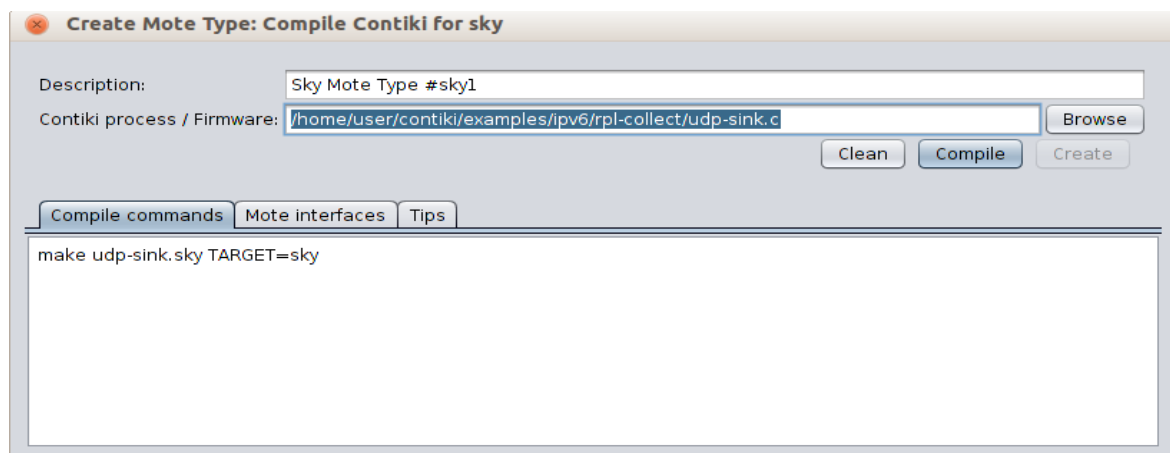
6. En la pestaña de motes se da clic en “Add motes”/”Create new mote type” y se selecciona en el menú desplegable entre los distintos tipos de motes que ofrece el simulador, para las simulaciones realizadas se escoge el mote tipo “sky”.

Figura 6 Paso 6 de implementación de simulación



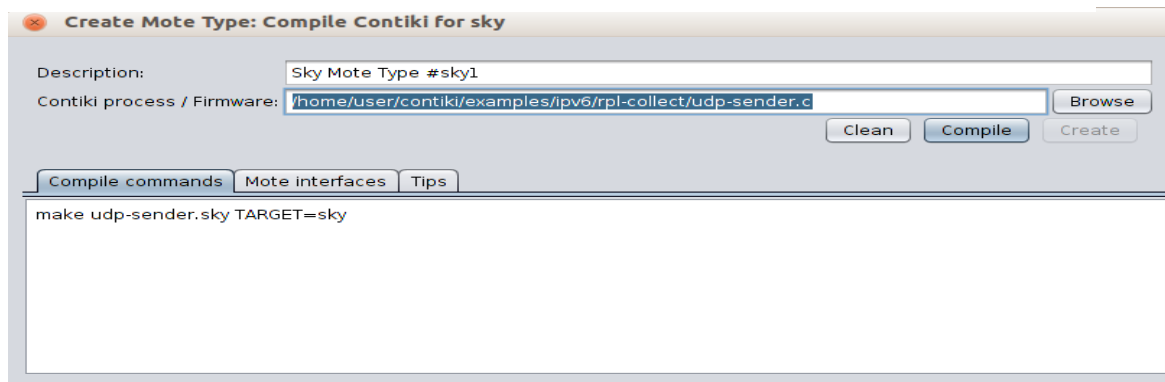
7. Se despliega un cuadro donde se debe dar la ruta de la aplicación que se usará como firmware de los nodos de la simulación, para las simulaciones realizadas se usa el aplicativo *udp-sender.c* que será el firmware de los nodos remitentes y *udp-sik.c* para el nodo raíz. En este paso se escoge el firmware para escoger el nodo raíz ubicado en la ruta */home/user/contiki/examples/ipv6/rpl-collect/udp-sink.c*.

Figura 7 Paso 7 de implementación de simulación



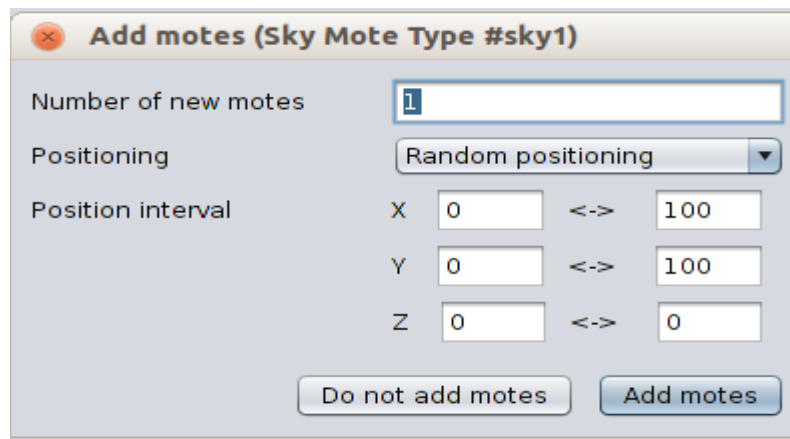
8. Se escoge el firmware para los nodos remitentes que está ubicado en la dirección `/home/user/contiki/examples/ipv6/rpl-collect/udp-sender.c`, se da clic en *compile* y luego en *create*.

Figura 8 Paso 8 de implementación de simulación



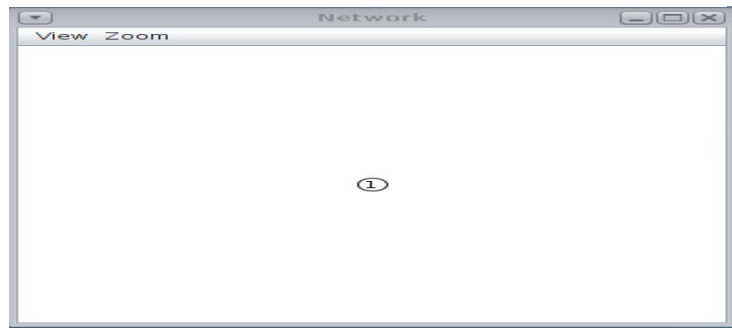
9. Una vez creado los dos tipos de motes, nuevamente en la pestaña de *motes* se selecciona por el primer tipo de mota para agregar una como nodo raíz, emerge un cuadro donde debo ingresar la cantidad de nuevas motas que se van a agregar y luego se da clic en el botón *add motes*.

Figura 9 Paso 9 de implementación de simulación



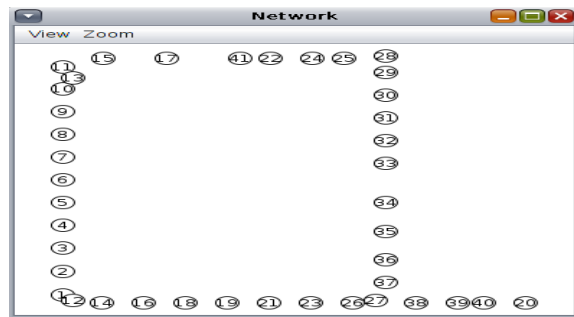
10. En el recuadro llamado *Network* se verá el nodo representado por un círculo

Figura 10 Paso 10 de implementación de simulación



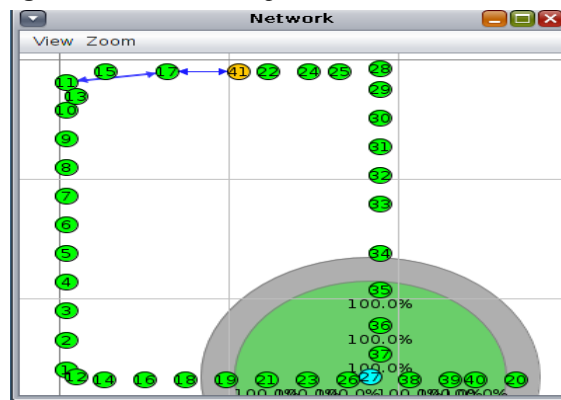
11. Posteriormente se agrega los nodos remitentes

Figura 11 Paso 11 de implementación de simulación



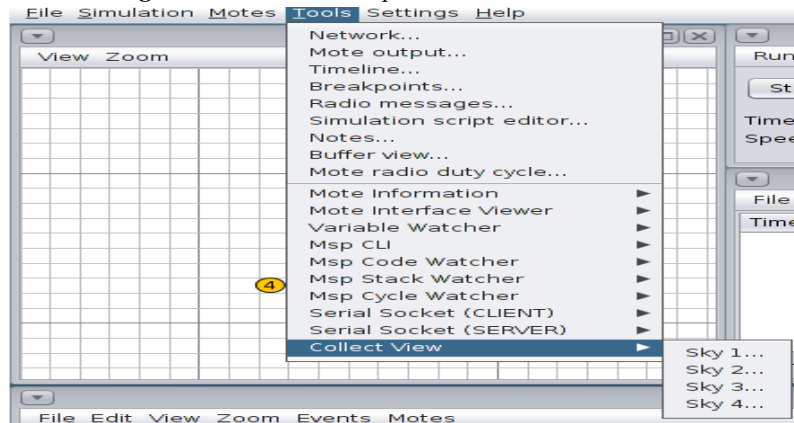
12. En el recuadro de *network* en la pestaña de *view* al dar clic se despliega un menú donde se selecciona las opciones *radio traffic* para ver la representación de las conexiones radiales entre los nodos, *mote type* para colorear los motes dependiendo del tipo, *radio environment* para ver la representación del alcance de radio y *background grid* para ver la cuadrícula de fondo.

Figura 12 Paso 12 de implementación de simulación



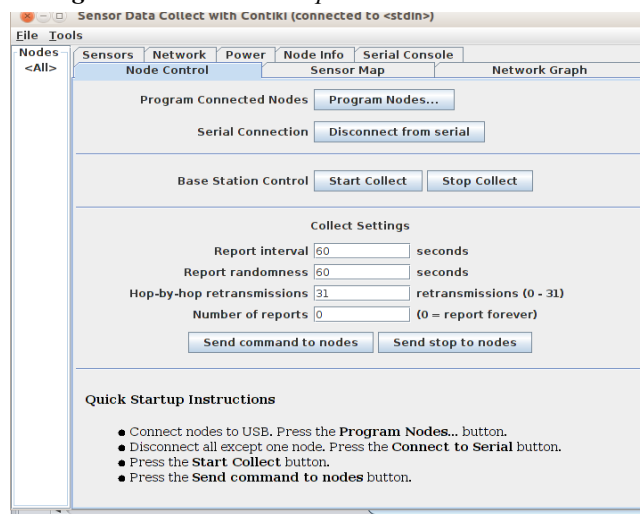
13. Luego en la pestaña *tools* se despliega un menú, donde se observan todas las herramientas con que cuenta el simulador *cooja*, en dicho menú seleccionamos *collect view* herramienta que permite monitorear las métricas de los nodos, se selecciona el nodo raíz el cual en conjunto con el firmware y el *collect view* se podrá observar las métricas de la red, si se selecciona un nodo distinto al nodo raíz la aplicación *collect view* no funciona de manera adecuada y no se podrá recolectar los datos.

Figura 13 Paso 13 de implementación de simulación



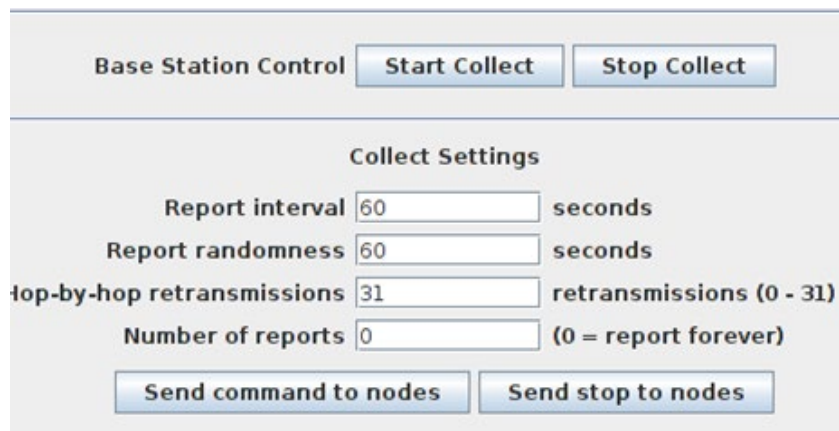
14. Una vez seleccionado *collect view* se despliega el siguiente cuadro.

Figura 14 Paso 14 de implementación de simulación



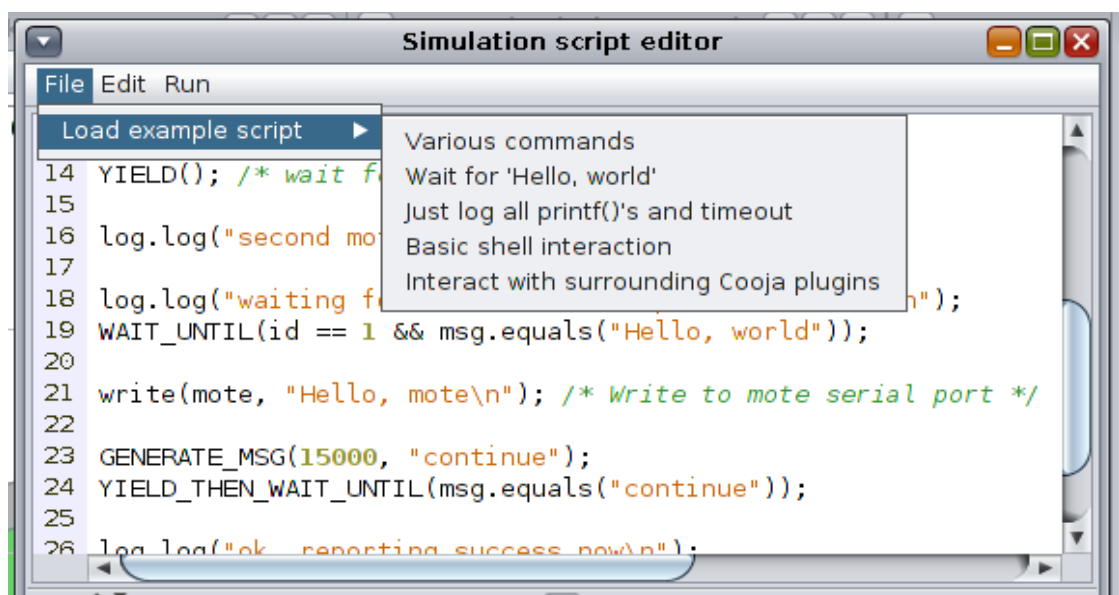
15. En la sección de *collect settings* se configura los parámetros de periodicidad con la que se monitorea a la red, se da clic en *send command* y luego en *start collect* para iniciar el monitoreo

Figura 15 Paso 15 de implementación de simulación



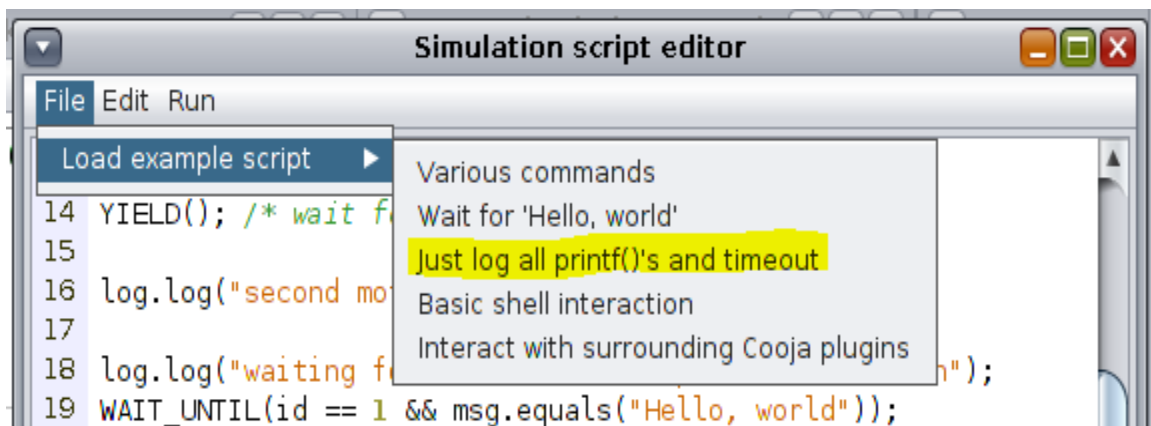
16. Para automatizar el tiempo de la simulación y detener automáticamente se usa una herramienta *simulation script editor*.

Figura 16 Paso 16 de implementación de simulación



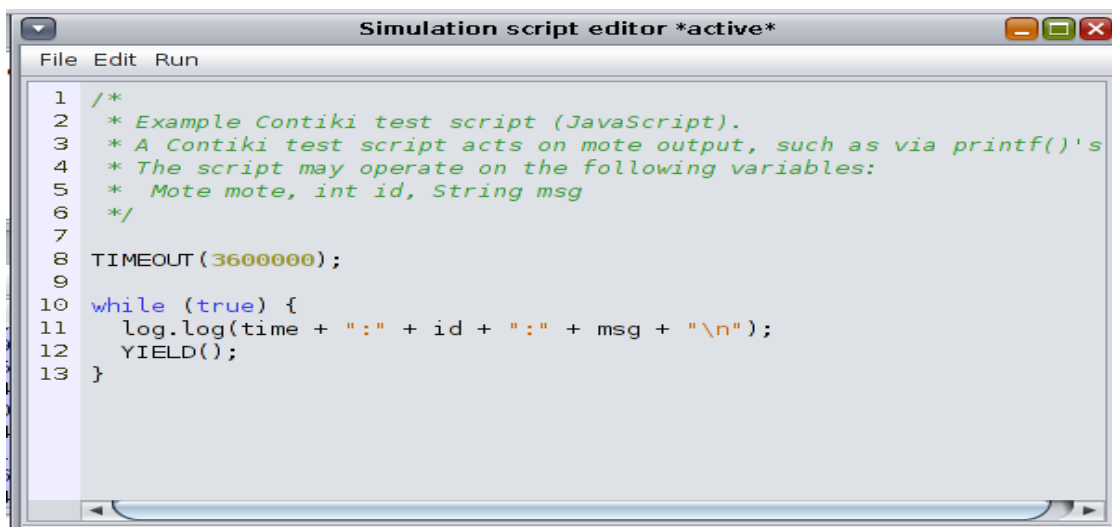
17. Seleccionar en la pestaña file seleccionamos load *example script* y se despliega un menú en el cual se selecciona el ejemplo *just log all printf() 's and timeout*.

Figura 17 Paso 17 de implementación de simulación



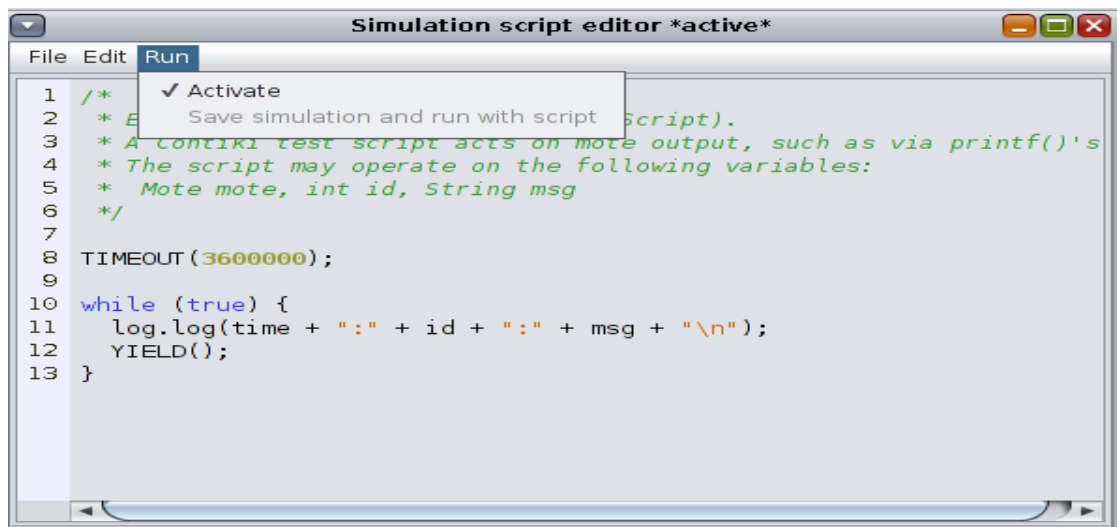
18. Luego se seleccionar el ejemplo se carga el siguiente script, en la función TIMEOUT () dentro de los paréntesis se escribe el tiempo en milisegundos que se desea que dure la simulación, para las simulaciones realizadas en este proyecto se escoge una duración de una hora lo que equivale a tres seiscientos mil milisegundos.

Figura 18 Paso 18 de implementación de simulación



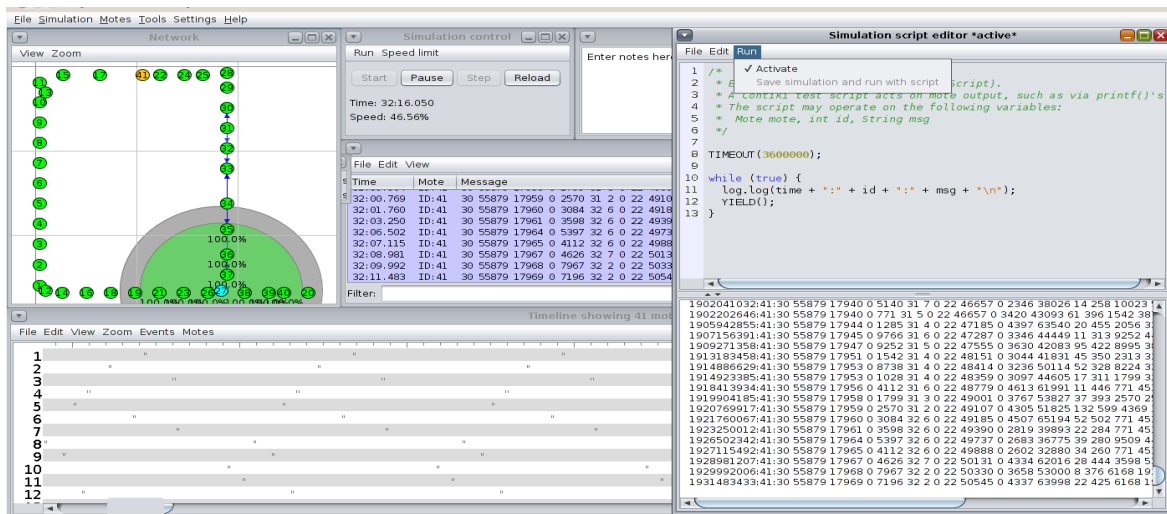
19. En la pestaña Run se despliega un menú donde se selecciona la opción *Activate* para activar el script.

Figura 19 Paso 19 de implementación de simulación



20. Una vez realizado todos los pasos anteriores en el cuadro de *simulation control* en el botón *start* dando clic en este se inicia la simulación.

Figura 20 Paso 20 de implementación de simulación



21. Cuando finaliza el tiempo de simulación en el aplicativo de *collect view* hay una pestaña en llamada *Node Info* en donde se observa todos los datos recolectados de la simulación, los cuales se copian en un archivo de texto para su posterior análisis .

Figura 21 Collect View

Node	Received	Dups	Lost	Hops	Rtmtrc	ETX	Churn	Beacon Interval	Reboots	CPU Power
1.1	34	0	0	6.000	1094.3...	59...	0	23 min, 31 sec	0	0.379
2.2	34	0	0	2.000	579.059	24...	0	23 min, 42 sec	0	0.340
3.3	33	0	1	6.000	1109.2...	58...	0	24 min, 03 sec	0	0.380
4.4	33	0	1	1.000	384.273	16...	0	23 min, 51 sec	0	0.442
5.5	31	0	2	7.000	1308.9...	67...	2	19 min, 15 sec	0	0.365
6.6	32	0	2	7.000	1269.6...	67...	2	19 min, 26 sec	0	0.358
7.7	33	0	0	7.000	1378.0...	76...	0	22 min, 05 sec	0	0.320
8.8	31	0	1	6.032	1245.2...	61...	1	20 min, 52 sec	0	0.372
9.9	33	0	0	1.000	433.818	16...	0	23 min, 45 sec	0	0.348
10.10	33	0	0	6.000	1113.4...	60...	0	22 min, 44 sec	0	0.381
11.11	33	0	0	1.000	384.576	16...	0	23 min, 37 sec	0	0.458
12.12	33	0	0	1.000	452.182	16...	0	23 min, 53 sec	0	0.361
13.13	33	0	1	6.000	1172.6...	62...	0	24 min, 06 sec	0	0.390
14.14	34	0	0	5.000	1008.1...	51...	0	23 min, 34 sec	0	0.391
15.15	34	0	0	2.000	588.618	24...	0	23 min, 42 sec	0	0.361
16.16	33	0	0	2.000	575.152	24...	0	23 min, 09 sec	0	0.370
17.17	33	0	0	2.000	513.636	24...	0	23 min, 53 sec	0	0.439
18.18	34	0	0	2.000	587.618	24...	0	24 min, 11 sec	0	0.363
19.19	34	0	0	3.000	661.500	32...	0	23 min, 32 sec	0	0.350
20.20	33	0	1	3.000	647.879	32...	0	24 min, 17 sec	0	0.411
21.21	34	0	0	4.000	838.212	42...	0	23 min, 53 sec	0	0.328
22.22	34	0	0	4.000	821.794	43...	0	23 min, 38 sec	0	0.410
23.23	30	0	3	5.000	958.967	51...	0	23 min, 33 sec	0	0.429
24.24	32	0	1	5.000	962.250	51...	0	23 min, 55 sec	0	0.398
25.25	34	0	0	6.000	1127.5...	60...	0	23 min, 13 sec	0	0.375
26.26	33	0	0	6.000	1102.6...	60...	0	23 min, 07 sec	0	0.374
27.27	33	0	1	6.000	1157.7...	60...	0	22 min, 43 sec	0	0.364
28.28	0	0	0	0.000	0.000	0.000	0		0	0.000
29.29										
30.30										
Avg	33.025	0.000	0.350	4.002	837.894	42...	0.150	23 min, 15 sec	0.000	0.379