World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Simulation of 6LowPAN Protocol in public Lightning management

Abdou Karim FAROTA * and Aminata WADE

Laboratory of Atmospheric and Ocean Sciences, Materials, Energy and Devices (LSAO-MED), Gaston Berger University of Saint-Louis, Senegal.

World Journal of Advanced Research and Reviews, 2023, 19(03), 1535–1552

Publication history: Received on 17 August 2023; revised on 27 September 2023; accepted on 29 September 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.19.3.1970

Abstract

Public lighting is all the means used to illuminate public spaces, roads and residential areas. It contributes to user safety and accounts for nearly a third of electricity consumption worldwide. The lamp lighting system was on/off. That is to say, either the street lamps are 100% on without taking into account the observed traffic density (much, little or no traffic), or they are all off. Faced with budget restrictions and rising energy costs, more and more local authorities are looking for a solution to reduce their electricity consumption related to street lighting.

We collected data in an RPL network of 10 nodes in random representation. A 6LoWPAN network of 30 nodes plus an additional node, representing the edge router is realized and simulated. The results obtained were presented, analysed and interpreted.

Keywords: 6LowPAN; WSN; Public lighting; RPL

1. Introduction

As a solution, low consumption is adopted with the change of the luminaire material by replacing sodium lamps with LEDs (Light Emitting Diode). White light sources promote better vision of the human eye. The response time of the eye is reduced, which becomes an asset for the road safety of drivers [1,2]. It should be noted that LEDs consume much less energy than sodium lamps. Discharge lamps can go down to a maximum of 50% of their rated power for HPS lamps and 40% for mercury. Below this power level they no operate. The first advantage of LEDs is that they can be lowered even further, with reductions of up to 90%. And this type of lighting can rise to 100% of its rated power instantly. After the advent of LEDs, we are witnessing a digitisation of lighting made possible by new information technologies, connected objects.

Throughout the world, the number of objects to be connected is spending millions and is constantly increasing. The Cisco Internet Business Solutions Group announced 15 billion IDOs in 2015 and forecasts 50 billion by 2020 [3]. The use of IPv6 in WSNs was considered unfeasible in advance. The low resources of these types of networks and the extra header size of IPv6 packets, giving a size of 1280 bytes for a packet to be transmitted, do not make the task any easier. Indeed, the Data Link and Physics layers of WSNs implement the IEEE 802.15.4 protocol, which is used in low-power networks. However, the draft IETF standard of the 6LoWPAN protocol changes this, by defining how IPv6 packets can be efficiently transmitted over IEEE 802.15.4 radio links. 6LoWPAN defines an adaptation layer for transporting IPv6 packets over low data rates, low power, small radio networks [4]. The 6LoWPAN adaptation layer is located between the Network layer and the Data Link layer. The 6LoWPAN protocol allows the fragmentation of IPv6 packets into 802.15.4 packets and reassembly of IPv6 packets upon reception.

---

* Corresponding author: Abdou Karim FAROTA

The main objective of this paper is to offer a sensor-based public lighting system for the reduction of energy consumption with real-time supervision of the equipment. Thus, the work will consist of five sections, the first of which is a study the communication protocols used in our WSN, namely the IEEE802.15.4 standard, the IPv6 protocol and the 6LoWPAN protocol. Afterwards, we will tackle the second section, we will present the simulation tools of the practical part : Contiki-Cooja and Proteus Professional 8.0. In section three will be devoted to the presentation of the results obtained in the different simulations as well as the analysis and interpretation of the results. And section four, we will present a model of intelligent street lighting system. Section Five led conclusion.

2. Communication protocols in WSN

The operation of wireless sensor networks requires communication protocols adapted to these characteristics. The minimum energy consumption, the main constraint of WCSNs, requires the use of specific protocols such as the 802.15.4 protocol. To make real-time and remote supervision possible, the IPv6 protocol makes the task easier with its use in RCSFs. Due to the often variable size of IPv6 packets, these packets are large for transmission over an 802.15.4 radio link. The creation of a new protocol is necessary, the 6LoWPAN protocol.

The 802.15.4 protocol was created by the IEEE. It defines the architecture of the Physical and Data Link layers. It is the basis of many other protocols such as Zigbee and 6LoWPAN. It is generally used in low resource networks (LRWPAN: Low Rate Wireless Power Area Network). It is an open protocol and the documentation is available on the Internet. The IEEE802.15.4 protocol is used in the short-range radio domain. Its general interest is its low energy consumption [5].

Channel sharing is achieved using multiple access carrier detection (CSMA) and acknowledgements are provided for reliability. Data Link Layer security is provided with 128-bit AES encryption. The payload of its physical layer is estimated to be up to 127 bytes [6].

The main features of the IEEE802.15.4 protocol are defined below:

- Carrier frequency: 2.4 GHz (general), 915 MHz for America, 868 MHz for Europe;
- Maximum data rate: 250 Kbps (for O-QPSK with DSSS), 40 Kbps, 20 Kbps (for BPSK);
- Maximum packet size: 127 bytes ;
- Short range (few tens of meters), low power consumption.

At the link layer, the 802.15.4 standard supports the implementation of point-to-point, star or tree networks.

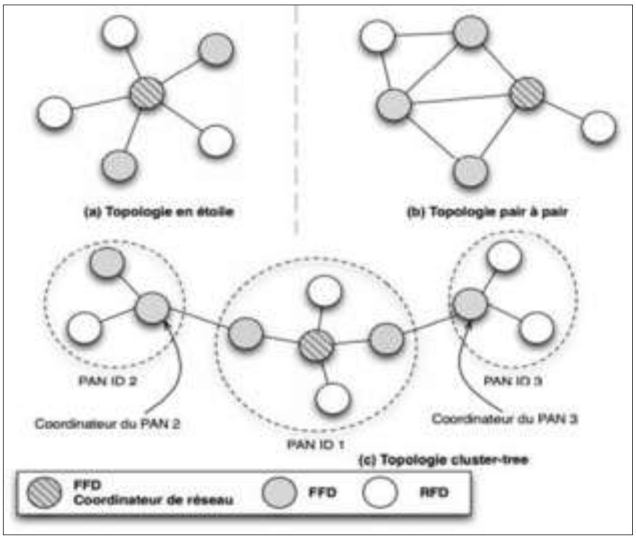


Figure 1 Existing topologies of an 802.15.4 network

The 802.15.4 network topologie is structured above (Fig.1). The FFD (Full Function Device) can perform three roles in a network : PAN coordinator, router or device connected to a sensor. The RFD (Reduced Function Device) is intended for simple applications (reporting the status of a sensor, controlling the activation of an actuator). It is considered an "end device" in the sense that it is not essential to the network. To communicate on the same network, an FFD (at least) and RFDs must use the same physical channel among those defined according to the chosen frequency band. The DFF can dialogue with RFDs and FFDs, while the RFD dialogues with an FFD only.

2.1. Protocol IPv6

Faced with address exhaustion and the explosion of routing tables, the IP version 6 protocol is emerging to overcome these problems. It has a wider address range than IPv4. We will see that IPv6 is not a radical change from IPv4, but an improvement requiring some protocol changes with better performance [7].

The address size of IPv6 is 2128 addresses unlike IPv4 which has 232 addresses. It comes with a larger number of nodes and simpler automatic address configuration. The scalability of multicast routing is improved by adding a scope field to multicast addresses. A new type of address called "anycast address" is defined and used to forward a packet to a group of nodes.

Some IPv4 header fields have been removed or made optional to reduce packet processing cost and limit IPv6 header bandwidth cost.

Changes in the way IP options are encoded allow for more efficient transmission, limit the length of less stringent options and provide great flexibility to introduce new options in the future.

A new feature is added to allow the labeling of packets belonging to a particular traffic flow for which the sender requests special treatment, such as quality of service or real-time service.

Extensions to support authentication, packet integrity, and data confidentiality are specified for IPv6.

2.2. Protocol 6LoWPAN

The 6LoWPAN protocol is a way to integrate IPv6 into low-resource networks such as 802.15.4 networks. It is represented as an adaptation layer located between the network layer and the data link layer by referring to the OSI (Open Systems Interconnection) model.

6LoWPAN also allows you to perform the configuration functions necessary to form and maintain an IPv6 subnet. The end-to-end use of infrastructure facilitates interoperability [6]. In the rest of this part we will see the architecture of a 6LoWPAN network (Fig. 2), the protocol stack, data routing and other subparts.

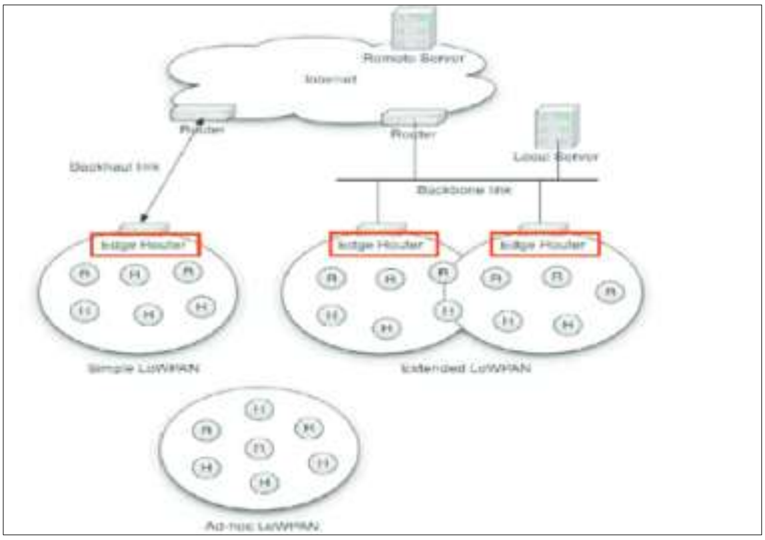


Figure 2 Global Architecture of a 6LoWPAN Network

2.3. Routing Protocol for Low-Power and Lossy Networks (RPL)

RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol for low-power, low-power networks such as LoWPAN networks [8]. RPL is a proactive distance vector (DV) designed to operate on the physical and MAC IEEE802.15.4 layers of RCSFs running IPv6. It targets node data collection by periodically sending measurements to a collection point and optimizes multipoint-to-point traffic. However, it does provide mechanisms that allow it to support both multipoint-to-point and point-to-point traffic [8].

The protocol is based on the topological concept of Directed Acyclic Graphs (DAGs). The DAG defines a tree structure, which specifies the default routes in the LLNs. The basic component of RPL is the DoDAG. The latter is a destination-oriented directed directed graph, rooted in a special node called the root DoDAG (Fig. 3).

The DoDAG root has the following properties [8]:

- It typically acts as a low-power, lossy edge (LBR) router;
- It represents the data well in the directed acyclic graph;
- It is typically the final destination node in the DoDAG, since it acts as a common transit point that connects the LLN with IPv6 networks;
- It has the ability to generate new DoDAG that descends to the leaf nodes.

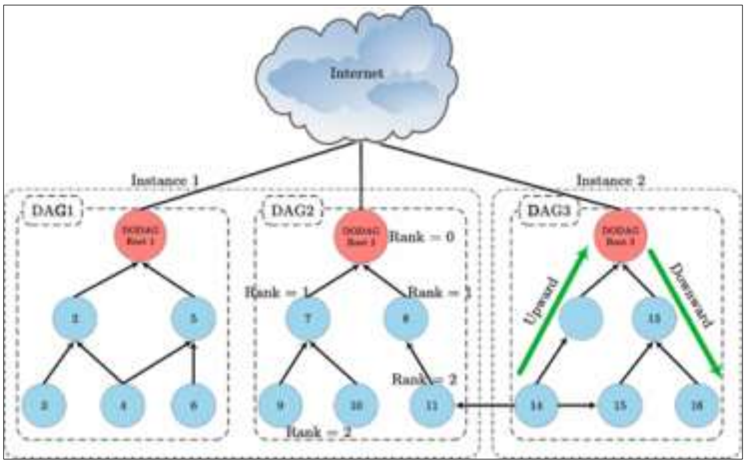


Figure 3 An RPL network with 3 DODAGs in 2 instances

3. Tools and results of simulation

3.1. Tools

For wireless sensor networks, testing their applications requires an environment that adapts to their characteristics. As part of our work, we will use Contiki (Fig. 4) which is accompanied by its simulator called Cooja [9].

The following properties have improved Contiki [9]:

- Internet standards: Contiki provides low-power Internet communications and supports IPv6 and IPv4 with low-power wireless standards: 6LoWPAN, RPL, CoAP;
- Rapid development: Application development with Contiki is quick and easy and programs are written in standard C language. These applications are emulated with the Cooja simulator before being injected into the real sensors. There is also the Instant Contiki environment which provides a development environment;
- A selection of hardware: Contiki works on a range of low-power wireless devices such as the Mica family and the Telos family;
- Active Community: Contiki is developed by a global team of developers with contributions from Atmel, Cisco, Redwire LLC, SAP, Thingsquare and many others, led by Adam Dunkels of Thingsquare;
- Open Source Software: Contiki can be used freely in both commercial and non-commercial systems;
- Commercial Support: Contiki provides both Contiki developer community support and commercial support.

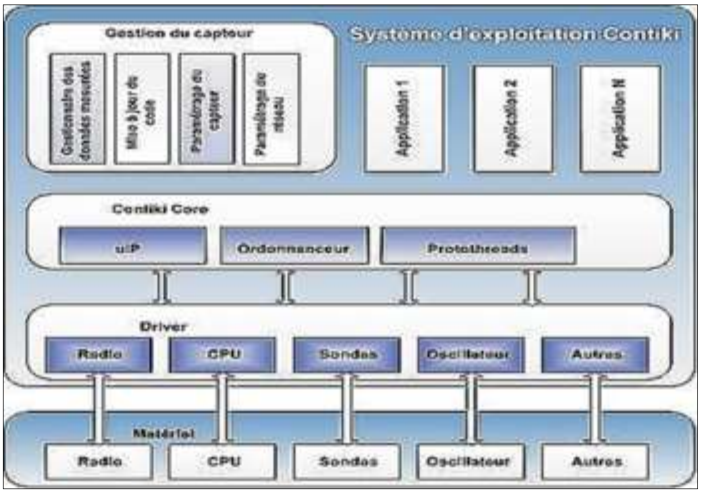


Figure 4 Contiki architecture

Cooja is a simulator coded in Java mainly built to simulate sensor networks using the Contiki micro OS. Cooja simulates sensor networks where each node can be of different type, not only by the embedded code, but also by the simulated hardware [10].

During simulations with Contiki we use the Sky-mote microcontroller. Its main features are as follows [10] :

- 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver; Interoperability with other IEEE 802.15.4 devices;
- 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash
- Integrated ADC, DAC, Power Supply Voltage Supervisor, and DMA controller; Integrated onboard antenna with a range of 50m indoor/outdoor 125m;
- Built-in humidity, temperature and light sensors;
- Ultra low current consumption; rapid awakening from sleep (<6 μ s);
- Hardware link layer encryption and authentication. Programming and data collection via USB;
- 16-pin expansion bracket and optional SMA antenna connector.

3.2. Results, analysis and interpretation of data collection

It is planned first simulation of a simple RPL network grouping a dozen nodes in order to know the performance of an RPL network. To know how a 6LoWPAN network works, we will do a simulation of a 6LoWPAN network with 31 nodes and finally we will do a last simulation on Proteus to show how the street lighting system works (Fig. 5).

We note that each node represents street lamp of our smart street lighting system. The duration of the first two simulations will be set at 30 minutes for each simulation.

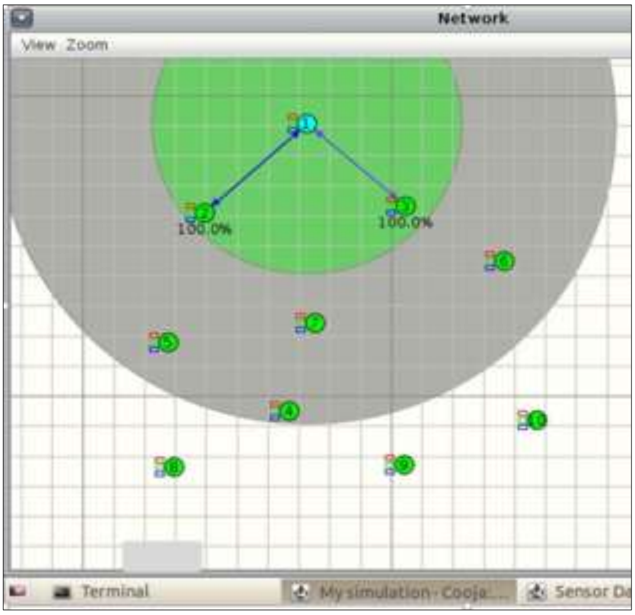


Figure 5 Wireless sensor network with RPL

Node 1 represents the root of the DoDAG (Fig. 6), it is the top of the DoDAG. Data collection is triggered on it, this is the collection point. The nodes are represented in a black dot with each its identifier (example: 1.0, 2.0 .... in black). All other nodes, regardless of their position in the network, seek to find a path to the root. To build the DoDAG, from the start of the simulation any node different from the root, looks for its parents and identifies with them as a child. Each node chooses a preferred parent and another in case the preferred parent is not accessible. This allows it to access the root faster. The blue arrows show the paths linking each node to its preferred parent to the root access. The values (1.0, 1.13, 1.38 in blue) represent the cost of a link between two nodes.

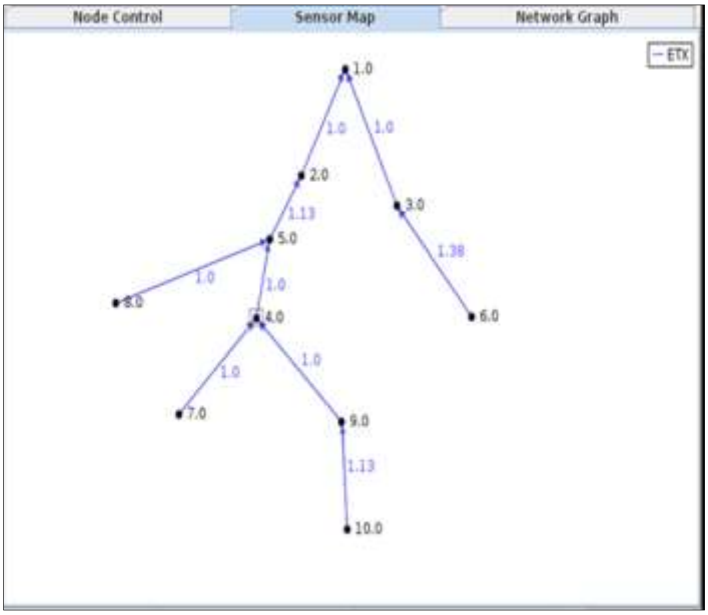


Figure 6 The DoDAG

We can see that nodes 2 and 3 are directly attached to root, this is explained by the fact that they are in the communication area of the root. The nodes are in the root interference zone, so their packets perform multiple hops to access the root.

The average power consumption of the grid is estimated in Figure (Fig. 7). The energy consumed by the network is centralized on four parameters: LPM (Low Power Mode), CPU (processor), radio listening and radio transmission, each represented by a color described in the legend on the figure at the bottom of the graph. It is represented in histogram with the number of the nodes as abscissa and the power consumed in mW as the ordinate. According to this figure, over a 30-minute simulation the highest average consumption is 14.8 mW corresponding to node 5.

This is understood by the fact that node 5 is the data relay of nodes 4, 7, 8, 9 and 10 (Fig. 6), so it consumes more memory, processor and radio.

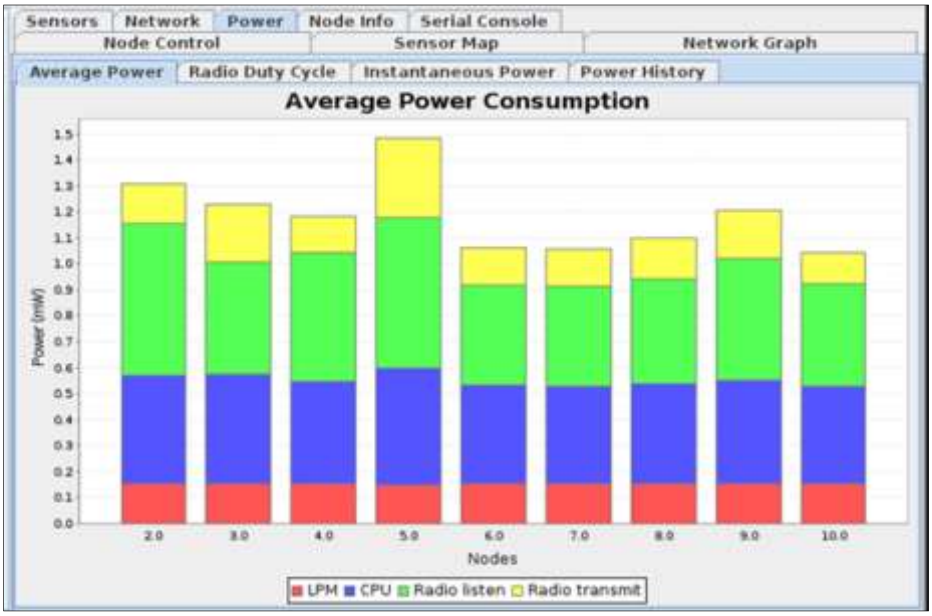


Figure 7 Average power consumption of nodes

Figure (Fig. 8) shows the histogram of the radio duty cycle of the nodes in average value. The color red represents radio listening and the blue color represents radio transmission of nodes. We notice that node 5 consumes more transmission and listening compared to other nodes with a rate of about 15.4%. Node 2 follows it with a higher rate compared to the remains of the nodes with a rate of 12.7%. Nodes 3, 4 and 9 also follow with rate respectively equal to : 11.4%, 10.9% and 11.3%.

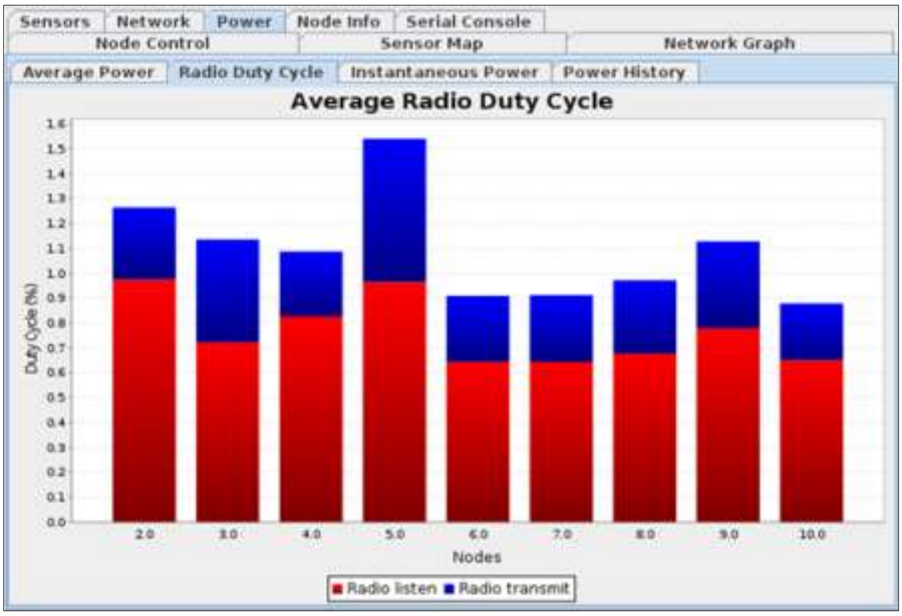


Figure 8 Histogram of node radio duty cycle in mean value

Radio consumption increases because of the wires of a node, because all packets from the child nodes must pass through the chosen parent nodes.

The number of hops recorded per node is represented by the histogram in Figure (Fig. 9). On each node, the average number of hops and the number of hops recently recorded is given. Node 10 records the most jumps in 30 minutes of simulation or more than 5 jumps. Nodes 7 and 8 record 4 hops. Referring to the acyclic graph, the number of hops for each node is equivalent to the number of radio links connecting the node to the root of the DoDAG.

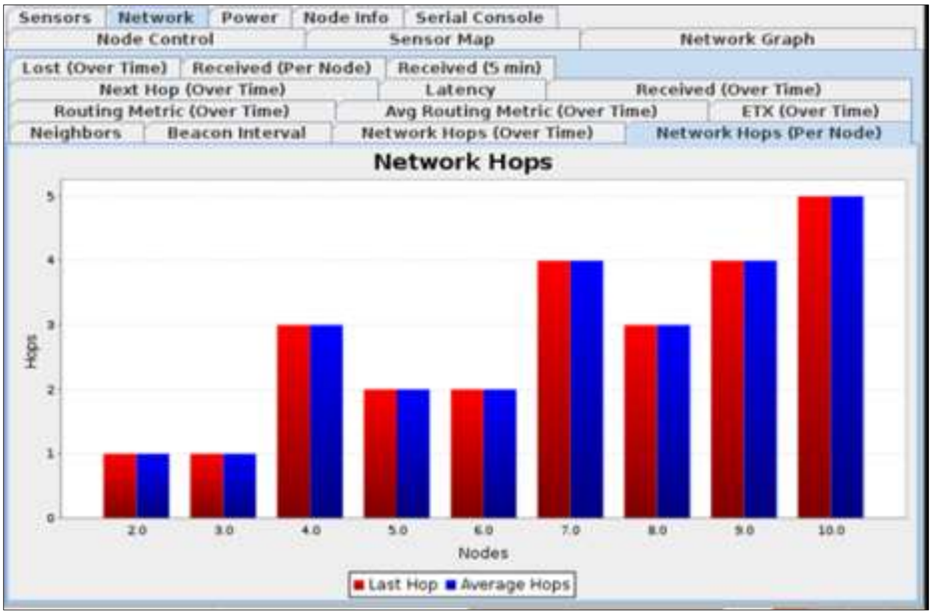


Figure 9 Histogram of the number of hops per node

The number of packets received per node is represented by the histogram in figure (Fig. 10). Nodes 2 and 3 receive more packets with 17 packets registered. The remaining 7 nodes receive 16 packets in 30 minutes of simulation.

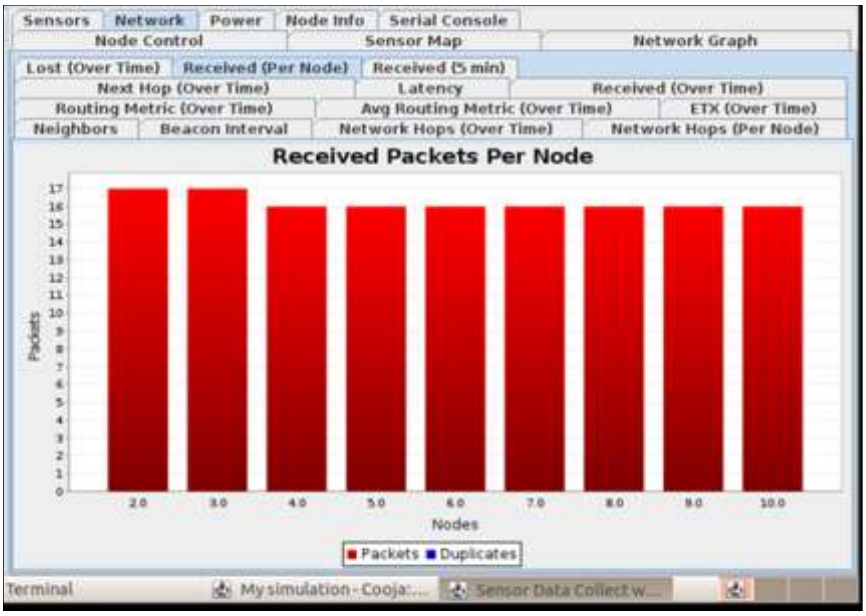
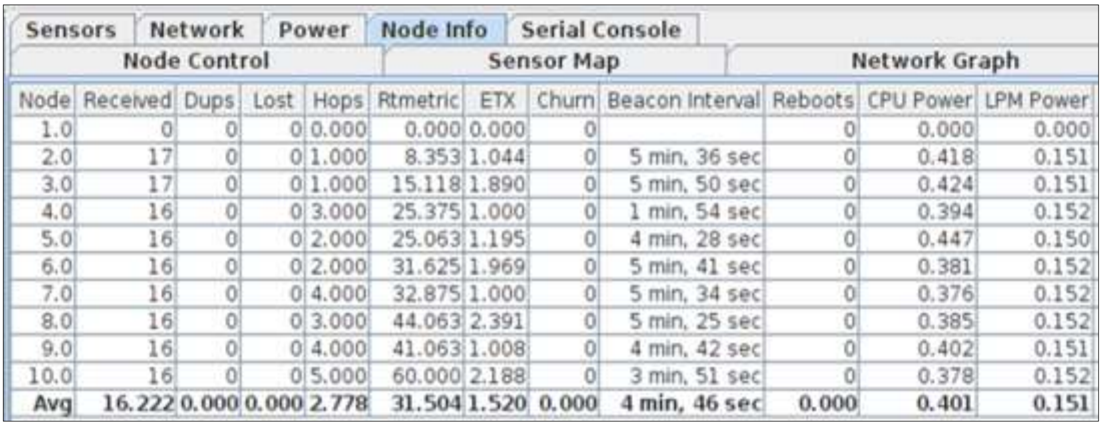


Figure 10 Histogram representing the number of packets received per node

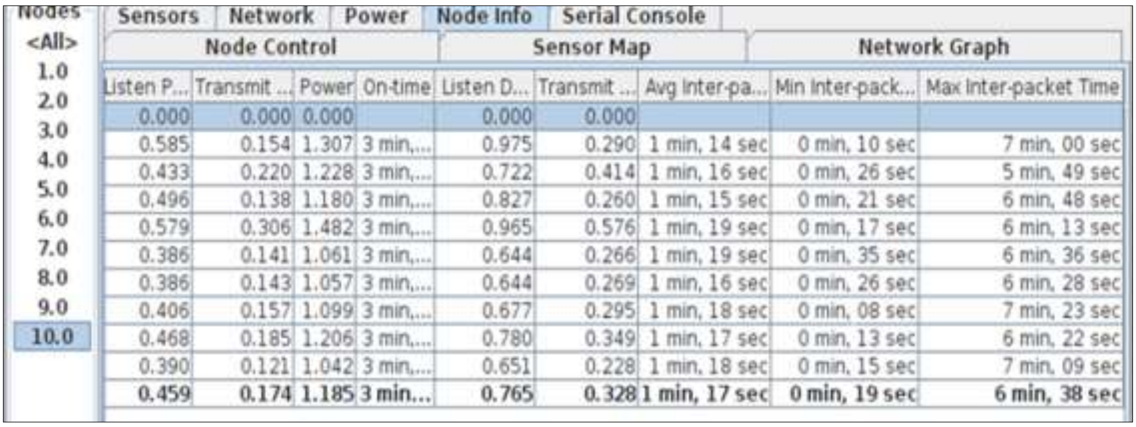
Table 1 and 2 show the variation in node-specific parameters and the equivalent in mean value for each parameter is given in bold on the last line.

Table 1 Nodes Informations



Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power
1.0	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000
2.0	17	0	0	1.000	8.353	1.044	0	5 min, 36 sec	0	0.418	0.151
3.0	17	0	0	1.000	15.118	1.890	0	5 min, 50 sec	0	0.424	0.151
4.0	16	0	0	3.000	25.375	1.000	0	1 min, 54 sec	0	0.394	0.152
5.0	16	0	0	2.000	25.063	1.195	0	4 min, 28 sec	0	0.447	0.150
6.0	16	0	0	2.000	31.625	1.969	0	5 min, 41 sec	0	0.381	0.152
7.0	16	0	0	4.000	32.875	1.000	0	5 min, 34 sec	0	0.376	0.152
8.0	16	0	0	3.000	44.063	2.391	0	5 min, 25 sec	0	0.385	0.152
9.0	16	0	0	4.000	41.063	1.008	0	4 min, 42 sec	0	0.402	0.151
10.0	16	0	0	5.000	60.000	2.188	0	3 min, 51 sec	0	0.378	0.152
Avg	**16.222**	**0.000**	**0.000**	**2.778**	**31.504**	**1.520**	**0.000**	**4 min, 46 sec**	**0.000**	**0.401**	**0.151**

Table 2 Nodes Informations



Listen P...	Transmit ...	Power	On-time	Listen D...	Transmit ...	Avg Inter-pa...	Min Inter-pack...	Max Inter-packet Time
0.000	0.000	0.000		0.000	0.000			
0.585	0.154	1.307	3 min,...	0.975	0.290	1 min, 14 sec	0 min, 10 sec	7 min, 00 sec
0.433	0.220	1.228	3 min,...	0.722	0.414	1 min, 16 sec	0 min, 26 sec	5 min, 49 sec
0.496	0.138	1.180	3 min,...	0.827	0.260	1 min, 15 sec	0 min, 21 sec	6 min, 48 sec
0.579	0.306	1.482	3 min,...	0.965	0.576	1 min, 19 sec	0 min, 17 sec	6 min, 13 sec
0.386	0.141	1.061	3 min,...	0.644	0.266	1 min, 19 sec	0 min, 35 sec	6 min, 36 sec
0.386	0.143	1.057	3 min,...	0.644	0.269	1 min, 16 sec	0 min, 26 sec	6 min, 28 sec
0.406	0.157	1.099	3 min,...	0.677	0.295	1 min, 18 sec	0 min, 08 sec	7 min, 23 sec
0.468	0.185	1.206	3 min,...	0.780	0.349	1 min, 17 sec	0 min, 13 sec	6 min, 22 sec
0.390	0.121	1.042	3 min,...	0.651	0.228	1 min, 18 sec	0 min, 15 sec	7 min, 09 sec
0.459	**0.174**	**1.185**	**3 min...**	**0.765**	**0.328**	**1 min, 17 sec**	**0 min, 19 sec**	**6 min, 38 sec**

The quality of the link of a hop between two nodes is evaluated using ETX (Fig. 11). These values are obtained at the receiving node and calculated using packet receive throughput. For node 5, the quality of the link of its hops varies between 1.0 and 2.0.

Figure 11 Cost of the next jump

The history of Node 5's power consumption over time is shown in figure (Fig. 12). It is represented over a time of 15 minutes. We notice that initially the power was equal to about 2.4 mW, over time this value decreases but not continuously. Each activity of the node corresponds to a power. The high consumption of node 5 is due to the number of neighbors it is the parent of. He is the relay of his two sons and his three grandsons, so normal that he consumes a lot of power.

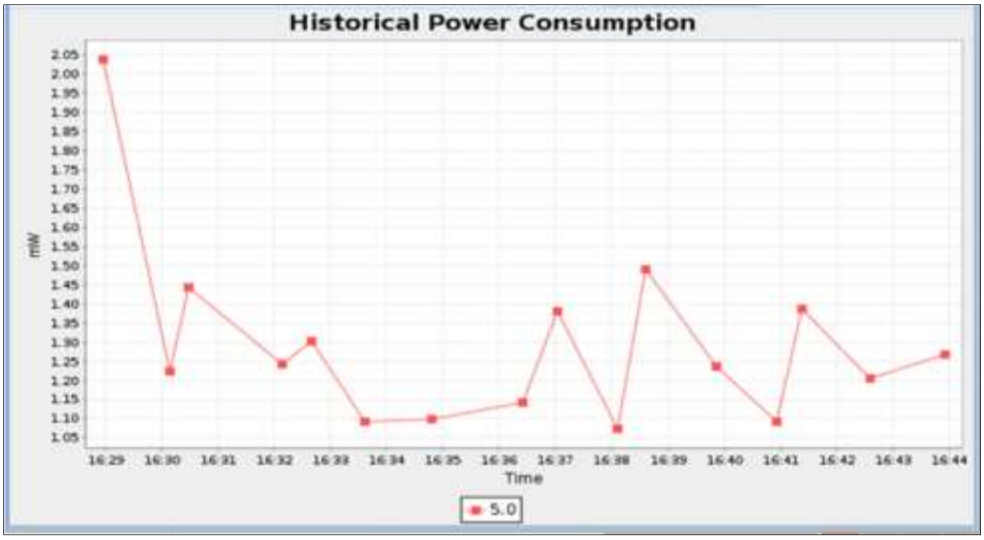


Figure 12 Node 5 Power Consumption

In Sky microcontroller we have two light sensors. The luminous intensity of the medium received by one sensor is shown in Figure 40. We notice that between 16:29 and almost at 16:39 the light intensity received decreases from 275 lux to almost 0 lux. Beyond 16:39 the intensity increases to 350 lux after one second and then decreases to 290 lux. This variation is due to the amount of ambient light in the medium where node 5 is exposed (Fig. 13).

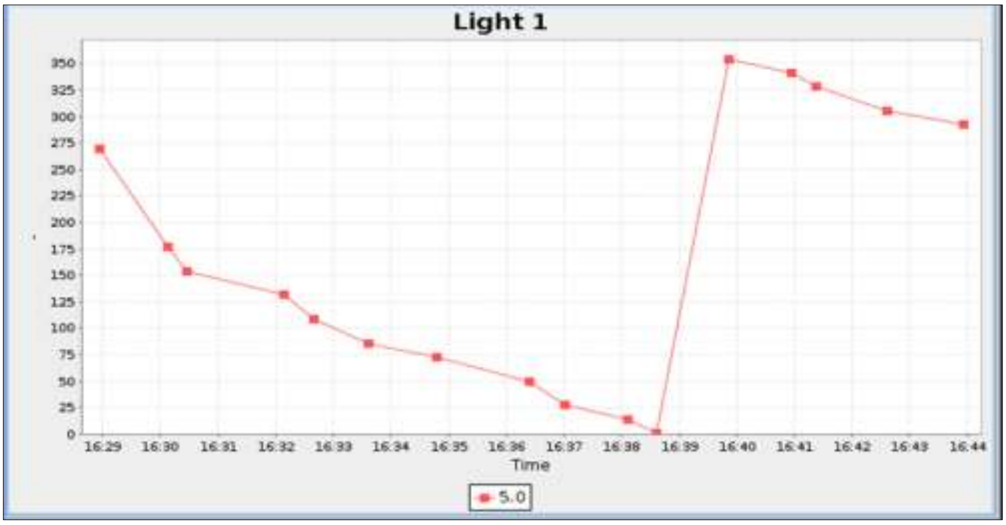


Figure 13 Luminous intensity received by light sensor 1

Figure (Fig. 14) shows the evolution of Node 5's battery consumption over time. In 15 minutes the node consume nearly 0.3 volts of its battery.

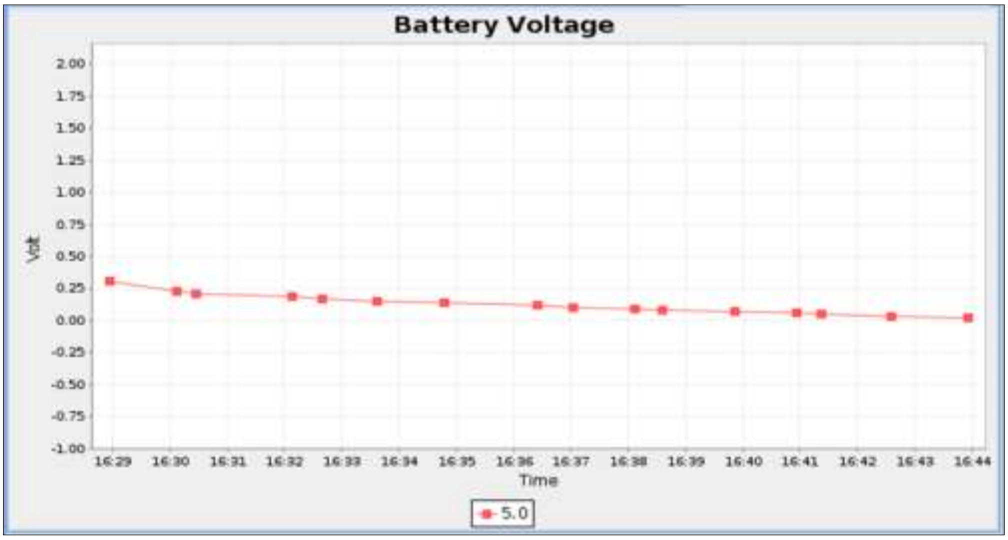


Figure 14 Node 5 battery consumption

3.3. Designing a 6LoWPAN Network in Contiki 2.7

To design a 6LoWPAN network, the first component to implement is the edge router. In the Cooja simulator, each component has its own path to use and simulate it, we specify that throughout the practical part the Sky mote node will be used.

In figure (Fig. 15), the edge router is represented by the green node with ID 1. For 6LoWPAN network, after setting up the edge router, you must add the nodes (hosts) of the network. In such network, we use the message transfer protocol is UDP, we will add UDP client nodes for networking.

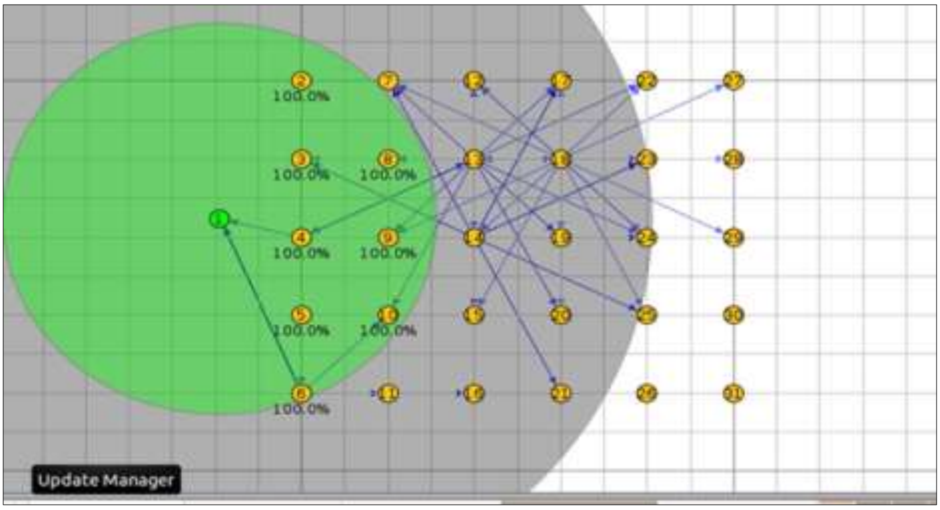
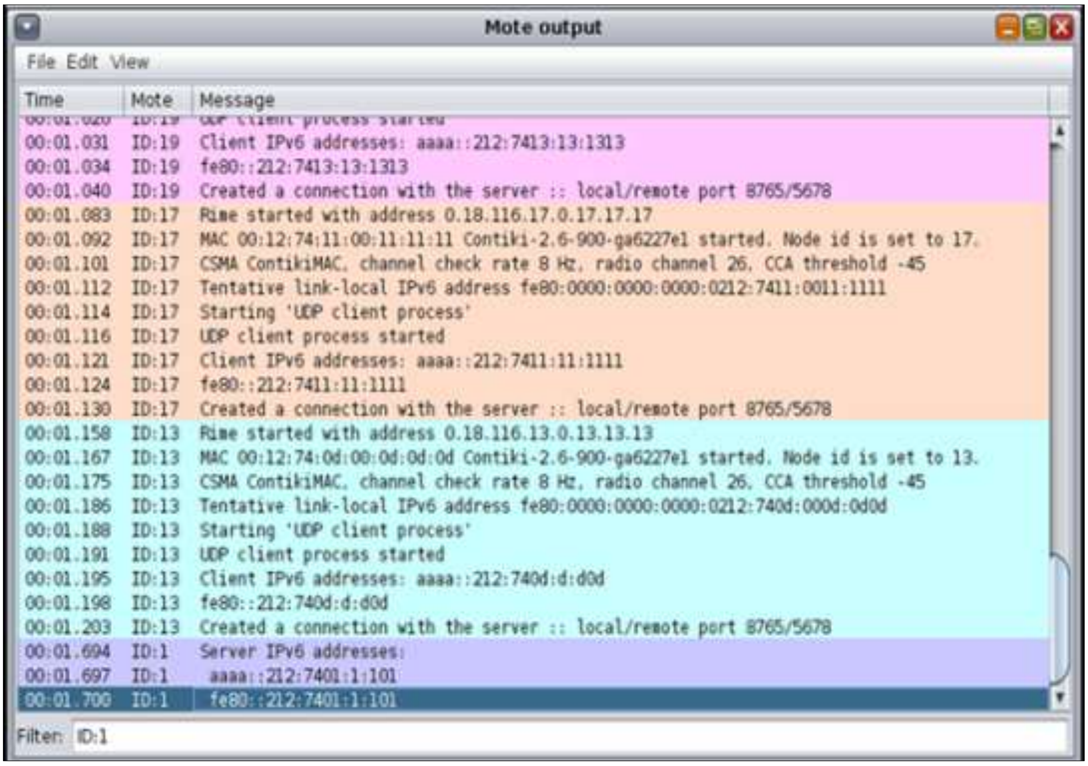


Figure 15 Architecture of 6LoWPAN network

The results of the simulation of the 6LoWPAN network will be structured by the following different screenshots beloved.

4. Results and discussion

As soon as the simulation starts, the Mote output window displays for each node: its communication time, the node identifier and the information provided. For the information displayed on the message part, the Rime address is given in the first line, then the MAC address in the second line, in the third line the default Contiki Mac mechanism, and on the fourth line the started process.

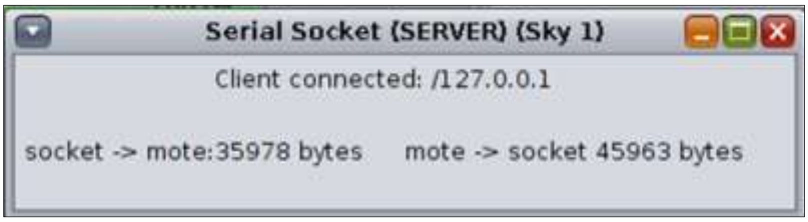


```
Time       Mote   Message
00:01.031  ID:19  Client IPv6 addresses: aaaa::212:7413:13:1313
00:01.034  ID:19  fe80::212:7413:13:1313
00:01.040  ID:19  Created a connection with the server :: local/remote port 8765/5678
00:01.083  ID:17  Rime started with address 0.18.116.17.0.17.17.17
00:01.092  ID:17  MAC 00:12:74:11:00:11:11:11 Contiki-2.6-900-ga6227e1 started. Node id is set to 17.
00:01.101  ID:17  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:01.112  ID:17  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7411:0011:1111
00:01.114  ID:17  Starting 'UDP client process'
00:01.116  ID:17  UDP client process started
00:01.121  ID:17  Client IPv6 addresses: aaaa::212:7411:11:1111
00:01.124  ID:17  fe80::212:7411:11:1111
00:01.130  ID:17  Created a connection with the server :: local/remote port 8765/5678
00:01.158  ID:13  Rime started with address 0.18.116.13.0.13.13.13
00:01.167  ID:13  MAC 00:12:74:0d:00:0d:0d:0d Contiki-2.6-900-ga6227e1 started. Node id is set to 13.
00:01.175  ID:13  CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
00:01.186  ID:13  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:740d:000d:0d0d
00:01.188  ID:13  Starting 'UDP client process'
00:01.191  ID:13  UDP client process started
00:01.195  ID:13  Client IPv6 addresses: aaaa::212:740d:d:d0d
00:01.198  ID:13  fe80::212:740d:d:d0d
00:01.203  ID:13  Created a connection with the server :: local/remote port 8765/5678
00:01.694  ID:1   Server IPv6 addresses:
00:01.697  ID:1    aaaa::212:7401:1:101
00:01.700  ID:1    fe80::212:7401:1:101
```

Figure 16 Node exits before edge router activation

In figure (Fig. 16), node 1 represents the edge router and acts as a server with global IPv6 address aaaa:212 :7401:1:101 and the rest of the nodes act as clients (ID13, 17, 19....). The client/server connection is established using port 8765/5678. So far, only the RPL network is active. To allow the RPL network to communicate with the Internet or an outside network, the edge router must be enabled.

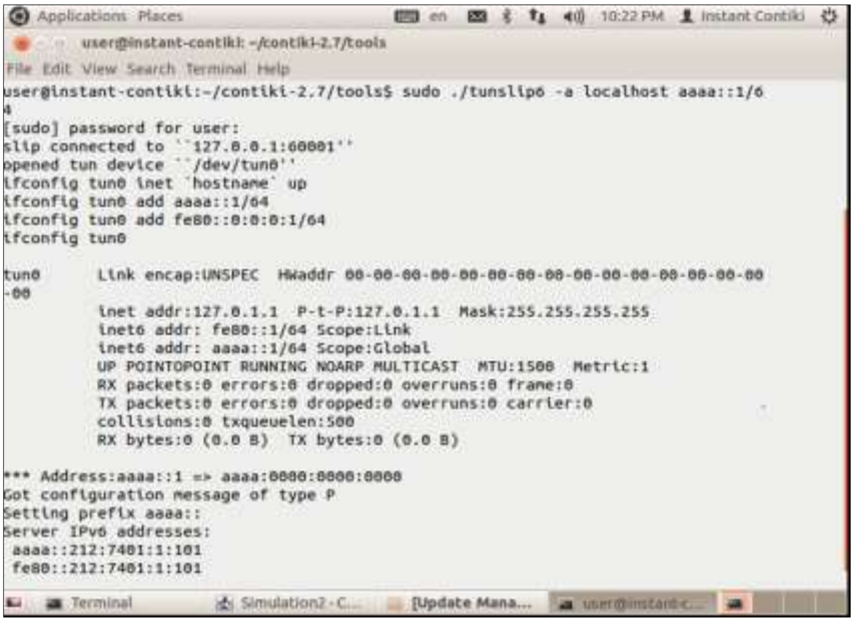
The results obtained will be shown in the following two figures (Fig. 17 and Fig. 18) :



```
Client connected: /127.0.0.1

socket -> mote:35978 bytes     mote -> socket 45963 bytes
```

Figure 17 Size of sockets sent and received by nodes through the serial port

The serial socket is used to connect the edge router to the Internet, in fact it maps the serial port of the edge router to the UDP port of the computer that houses the Cooja simulator. The edge router is connected to the machine's local server.



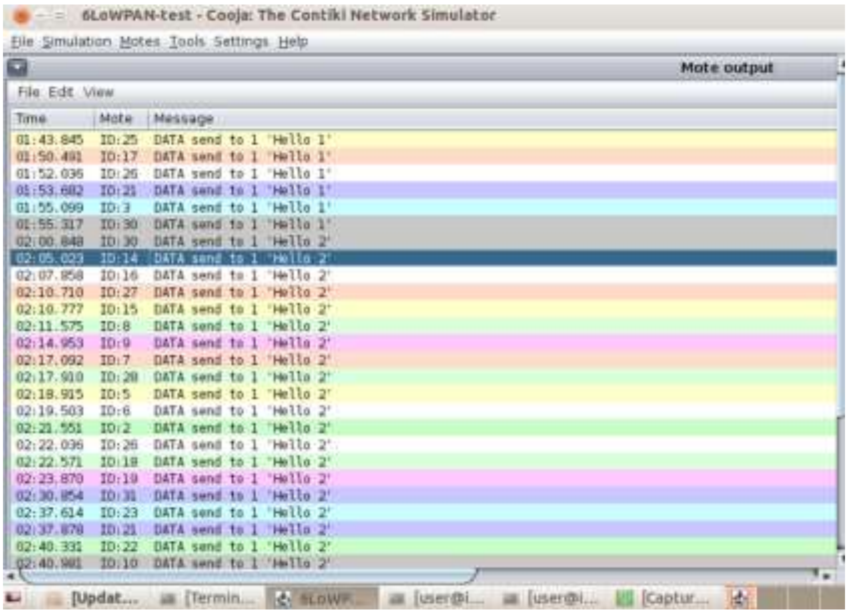
```
user@instant-contiki:~/contiki-2.7/tools$ sudo ./tunslip6 -a localhost aaaa::1/6
4
[sudo] password for user:
slip connected to ``127.0.0.1:60001''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` up
ifconfig tun0 add aaaa::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

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

Figure 18 Connecting the RPL network to the local machine

The edge router is the intermediary between the RPL network and the external network, to allow communication between these two networks, a "tunnelling" is applied on the router to allow the transition of data from the RPL network to the external network.

For this, Contiki provides tunslip accessible on tunslip6.c in contiki-2.7/tools. Tunslip is used to link IP traffic between a node to another element of the network (for example, the edge router). It creates a virtual interface (tun) on the host side and uses the Serial Line Internet Protocol (SLIP) protocol to encapsulate IP traffic to the other side of the serial line.

Figure (Fig. 19) shows "Hello" messages being sent from each node to the edge router. In fact, each node sends "Hello" message to inform the router that it is part of the network. The router assigns each node an IPv6 address and each node can then send this data to the Internet through the edge router.



```
Time        Mote    Message
01:43.845   ID:25   DATA send to 1 'Hello 1'
01:50.491   ID:17   DATA send to 1 'Hello 1'
01:52.036   ID:26   DATA send to 1 'Hello 1'
01:53.682   ID:21   DATA send to 1 'Hello 1'
01:55.099   ID:3    DATA send to 1 'Hello 1'
01:55.317   ID:30   DATA send to 1 'Hello 1'
02:00.848   ID:30   DATA send to 1 'Hello 2'
02:05.023   ID:14   DATA send to 1 'Hello 2'
02:07.858   ID:16   DATA send to 1 'Hello 2'
02:10.710   ID:27   DATA send to 1 'Hello 2'
02:10.777   ID:15   DATA send to 1 'Hello 2'
02:11.575   ID:8    DATA send to 1 'Hello 2'
02:14.953   ID:9    DATA send to 1 'Hello 2'
02:17.092   ID:7    DATA send to 1 'Hello 2'
02:17.910   ID:28   DATA send to 1 'Hello 2'
02:18.915   ID:5    DATA send to 1 'Hello 2'
02:19.503   ID:6    DATA send to 1 'Hello 2'
02:21.551   ID:2    DATA send to 1 'Hello 2'
02:22.036   ID:26   DATA send to 1 'Hello 2'
02:22.571   ID:18   DATA send to 1 'Hello 2'
02:23.870   ID:19   DATA send to 1 'Hello 2'
02:30.854   ID:31   DATA send to 1 'Hello 2'
02:37.614   ID:23   DATA send to 1 'Hello 2'
02:37.878   ID:21   DATA send to 1 'Hello 2'
02:40.331   ID:22   DATA send to 1 'Hello 2'
02:40.981   ID:10   DATA send to 1 'Hello 2'
```

Figure 19 Recognizing nodes at the edge router

Figure (Fig. 20) shows the radio and LED events for each node. For each node, we have the glow of its three LEDs lit corresponding to the colors blue, red and green and the radio activated correspond to the black dots.

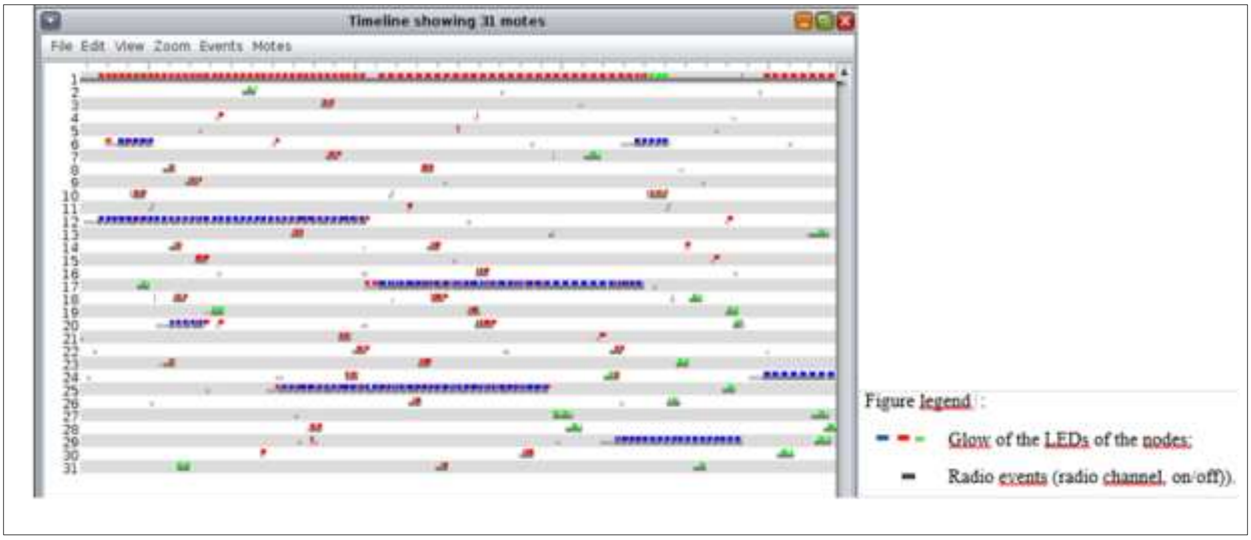
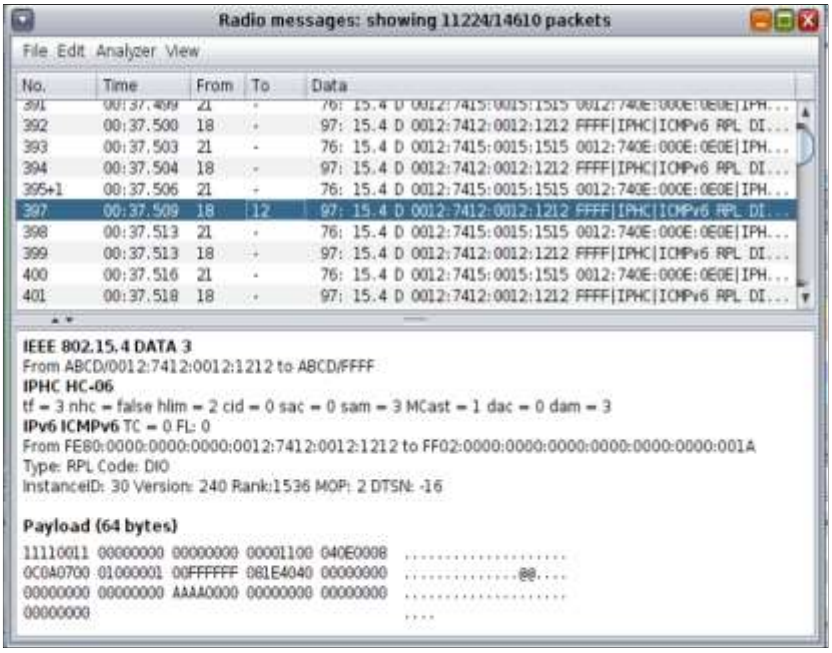


Figure 20 Radio events and node LEDs

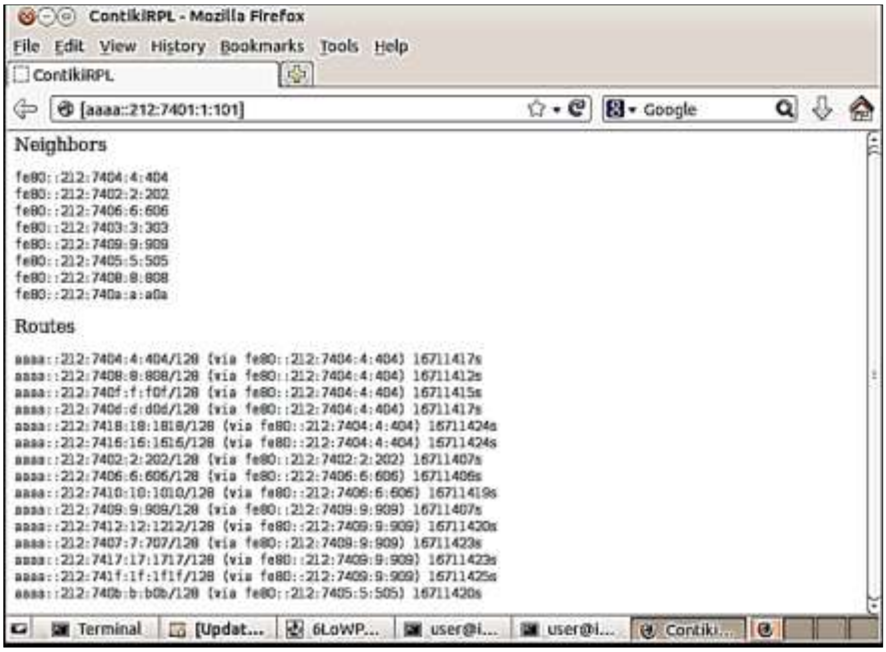
The analysis of 6LoWPAN packet radio messages is shown in figure (Fig.21). Message scanning is responsible for capturing packet traffic received in the network. For each packet, just click on it and see the packet information displayed: its source address and destination layer L2 and layer L3 are given as well as the value of each field of the packet. In packet 281, the RPL message is encoded in DIO (DODAC Information Object).



```
No.     Time       From  To   Data
391     00:37.499  21    -    76: 15.4 D 0012:7415:0015:1515 0012:740E:000E:0E0E|IPH...
392     00:37.500  18    -    97: 15.4 D 0012:7412:0012:1212 FFFF|IPHC|ICMPv6 RPL DI...
393     00:37.503  21    -    76: 15.4 D 0012:7415:0015:1515 0012:740E:000E:0E0E|IPH...
394     00:37.504  18    -    97: 15.4 D 0012:7412:0012:1212 FFFF|IPHC|ICMPv6 RPL DI...
395+1   00:37.506  21    -    76: 15.4 D 0012:7415:0015:1515 0012:740E:000E:0E0E|IPH...
397     00:37.509  18    12   97: 15.4 D 0012:7412:0012:1212 FFFF|IPHC|ICMPv6 RPL DI...
398     00:37.513  21    -    76: 15.4 D 0012:7415:0015:1515 0012:740E:000E:0E0E|IPH...
399     00:37.513  18    -    97: 15.4 D 0012:7412:0012:1212 FFFF|IPHC|ICMPv6 RPL DI...
400     00:37.516  21    -    76: 15.4 D 0012:7415:0015:1515 0012:740E:000E:0E0E|IPH...
401     00:37.518  18    -    97: 15.4 D 0012:7412:0012:1212 FFFF|IPHC|ICMPv6 RPL DI...

IEEE 802.15.4 DATA 3
From ABCD/0012:7412:0012:1212 to ABCD/FFFF
IPHC HC-06
tf = 3 nhc = false hlim = 2 cid = 0 sac = 0 sam = 3 MCast = 1 dac = 0 dam = 3
IPv6 ICMPv6 TC = 0 FL: 0
From FE80:0000:0000:0000:0012:7412:0012:1212 to FF02:0000:0000:0000:0000:0000:0000:001A
Type: RPL Code: DIO
InstanceID: 30 Version: 240 Rank:1536 MOP: 2 DTSN: -16

Payload (64 bytes)
11110011 00000000 00000000 00001100 040E0008  ....................
0C0A0700 01000001 00FFFFFF 081E4040 00000000  ..............@@....
00000000 00000000 AAAA0000 00000000 00000000  ....................
00000000                                      ....
```

Figure 21 Network packet scanning

Figure (Fig. 22) shows the layout of the router's routing table, which is accessible on the router's web server itself with access to the URL: [aaaa :: 212:7401:1:101] entered on the Contiki browser tab. Since in a 6LoWPAN network, all nodes have access to the Internet only through the edge router, a packet sent from one node to the Internet performs intermediate hops (two or more) if the node is not in the communication area of the router.



Neighbors

```
fe80::212:7404:4:404
fe80::212:7402:2:202
fe80::212:7406:6:606
fe80::212:7403:3:303
fe80::212:7409:9:909
fe80::212:7405:5:505
fe80::212:7408:8:808
fe80::212:740a:a:a0a
```

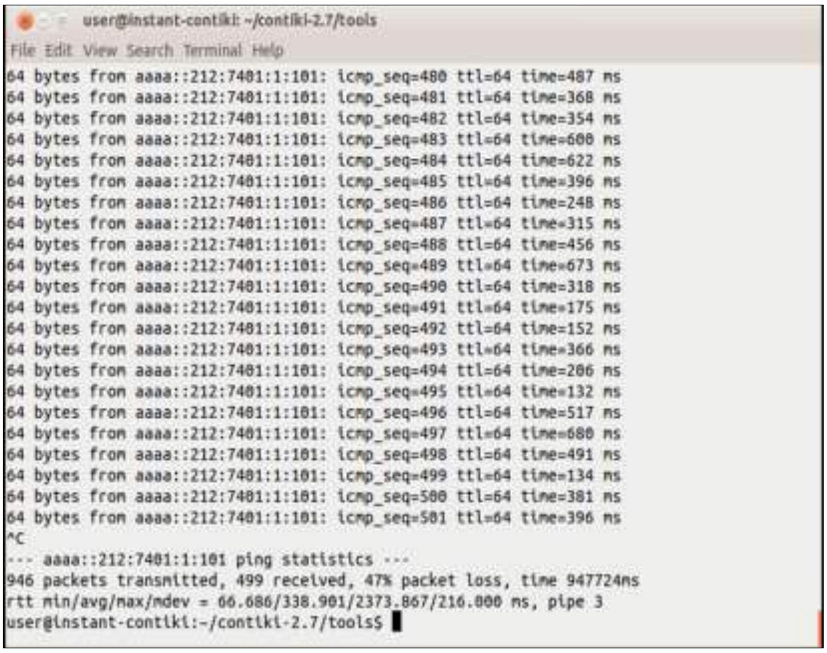
Routes

```
aaaa::212:7404:4:404/128 (via fe80::212:7404:4:404) 16711417s
aaaa::212:7408:8:808/128 (via fe80::212:7404:4:404) 16711412s
aaaa::212:740f:f:f0f/128 (via fe80::212:7404:4:404) 16711415s
aaaa::212:740d:d:d0d/128 (via fe80::212:7404:4:404) 16711417s
aaaa::212:7418:18:1818/128 (via fe80::212:7404:4:404) 16711424s
aaaa::212:7416:16:1616/128 (via fe80::212:7404:4:404) 16711424s
aaaa::212:7402:2:202/128 (via fe80::212:7402:2:202) 16711407s
aaaa::212:7406:6:606/128 (via fe80::212:7406:6:606) 16711406s
aaaa::212:7410:10:1010/128 (via fe80::212:7406:6:606) 16711419s
aaaa::212:7409:9:909/128 (via fe80::212:7409:9:909) 16711407s
aaaa::212:7412:12:1212/128 (via fe80::212:7409:9:909) 16711420s
aaaa::212:7407:7:707/128 (via fe80::212:7409:9:909) 16711423s
aaaa::212:7417:17:1717/128 (via fe80::212:7409:9:909) 16711423s
aaaa::212:741f:1f:1f1f/128 (via fe80::212:7409:9:909) 16711425s
aaaa::212:740b:b:b0b/128 (via fe80::212:7405:5:505) 16711420s
```

Figure 22 Edge router routing table

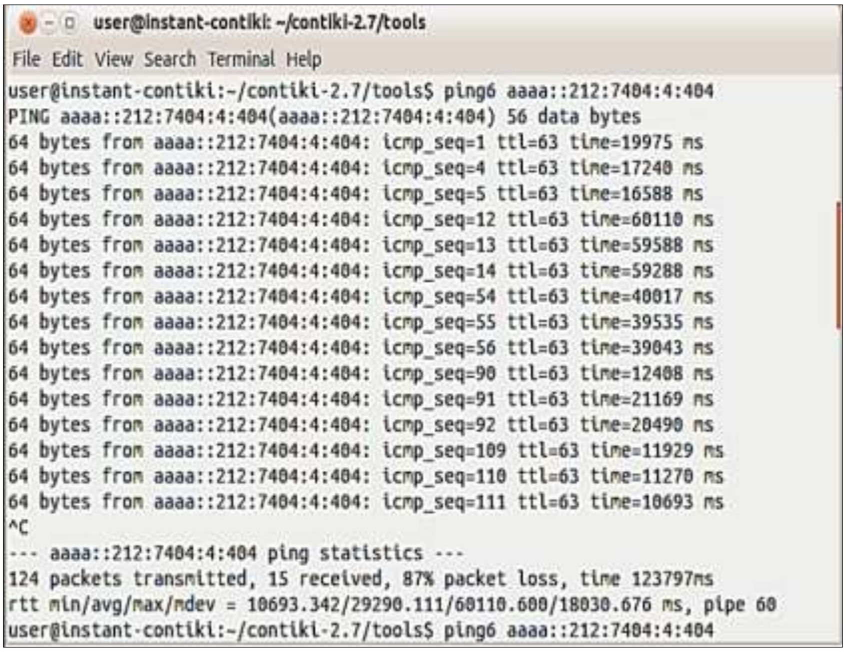
Here we see that only 8 nodes have direct access to the edge router whose address list is designated by Neighbor. This means that all routes of packets sent by a node outside the Neighbor list necessarily pass through these nodes.

Broadcasting in an IP network is used to verify the connectivity of an edge. In a 6LoWPAN network since the edge router passes all the data from the network to the external networks, so it is necessary to check its connectivity with the rest of the network. In an IPv6 network, to check the connectivity of the router, we use ping6. The ping6 command uses ICMPv6 ECHO_REQUEST to verify network connectivity. Its syntax is : ping6 <hostname>.



```
64 bytes from aaaa::212:7401:1:101: icmp_seq=480 ttl=64 time=487 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=481 ttl=64 time=368 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=482 ttl=64 time=354 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=483 ttl=64 time=608 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=484 ttl=64 time=622 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=485 ttl=64 time=396 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=486 ttl=64 time=248 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=487 ttl=64 time=315 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=488 ttl=64 time=456 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=489 ttl=64 time=673 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=490 ttl=64 time=318 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=491 ttl=64 time=175 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=492 ttl=64 time=152 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=493 ttl=64 time=366 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=494 ttl=64 time=206 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=495 ttl=64 time=132 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=496 ttl=64 time=517 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=497 ttl=64 time=680 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=498 ttl=64 time=491 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=499 ttl=64 time=134 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=500 ttl=64 time=381 ms
64 bytes from aaaa::212:7401:1:101: icmp_seq=501 ttl=64 time=396 ms
^C
--- aaaa::212:7401:1:101 ping statistics ---
946 packets transmitted, 499 received, 47% packet loss, time 947724ms
rtt min/avg/max/mdev = 66.686/338.901/2373.867/216.000 ms, pipe 3
user@instant-contiki:~/contiki-2.7/tools$
```

Figure 23 The edge router Braodcasting



```
user@instant-contiki:~/contiki-2.7/tools$ ping6 aaaa::212:7404:4:404
PING aaaa::212:7404:4:404(aaaa::212:7404:4:404) 56 data bytes
64 bytes from aaaa::212:7404:4:404: icmp_seq=1 ttl=63 time=19975 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=4 ttl=63 time=17240 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=5 ttl=63 time=16588 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=12 ttl=63 time=60110 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=13 ttl=63 time=59588 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=14 ttl=63 time=59288 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=54 ttl=63 time=40017 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=55 ttl=63 time=39535 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=56 ttl=63 time=39043 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=90 ttl=63 time=12408 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=91 ttl=63 time=21169 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=92 ttl=63 time=20490 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=109 ttl=63 time=11929 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=110 ttl=63 time=11270 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=111 ttl=63 time=10693 ms
^C
--- aaaa::212:7404:4:404 ping statistics ---
124 packets transmitted, 15 received, 87% packet loss, time 123797ms
rtt min/avg/max/mdev = 10693.342/29290.111/60110.600/18030.676 ms, pipe 60
user@instant-contiki:~/contiki-2.7/tools$ ping6 aaaa::212:7404:4:404
```

Figure 24 Node Broadcasting

In Contiki, you have to open a new terminal and write ping6 aaaa::212:7404:4:404 and validate on the enter key on the keyboard. Figure (Fig. 24) shows the result if the network is connected. The ping sent to the router is 56 bytes and the router responds to the ping by sending icmp messages of size 64 bytes at a time interval equal to 63.

4.1. Application to intelligent street lighting system

A smart street lighting system using 6LoWPAN is structured in Figure (Fig. 25). The edge router and lights communicate via radio. Any lamp that is out of range of the antenna of the radio module of the edge router, relays its information to it through the lamps closer to the router than it. The user connects to the edge router to access the information sent by all the lamps and act accordingly. Data processing tools are installed on the machine to allow the analysis and interpretation of the data received from the lamps. For each data received, it will be possible to know to which lamp it belongs and what is happening on it. The data processing allows the use to be able to interact with the network of lamps and know the situation on the network equipment.

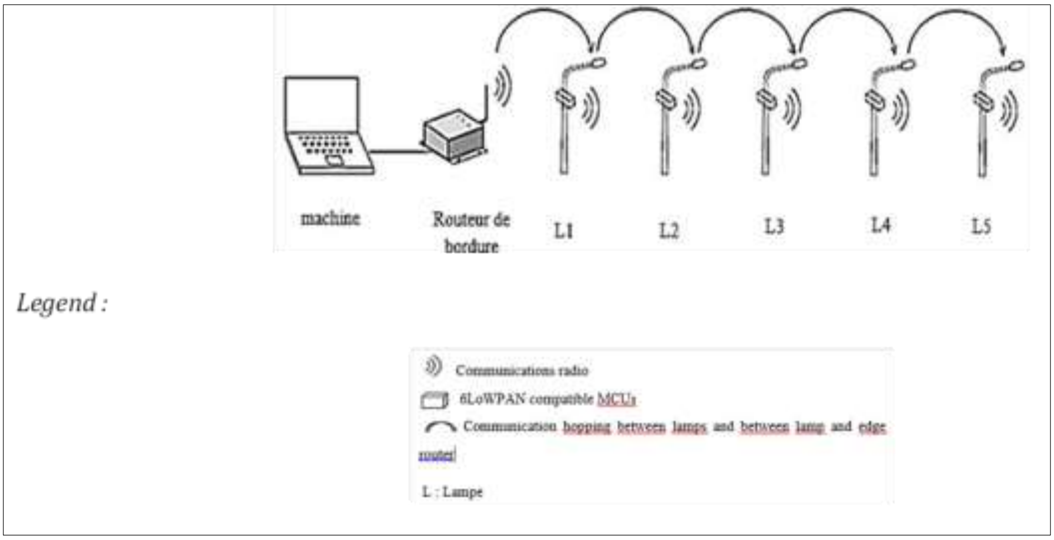


Figure 25 Street lighting system using 6LoWPAN

The Proteus software (Proteus Professional 8.0) has microcontrollers (arduino board) and sensors with which we can design an intelligent street lighting system. Proteus is used to show how our lighting system works (Fig. 26).

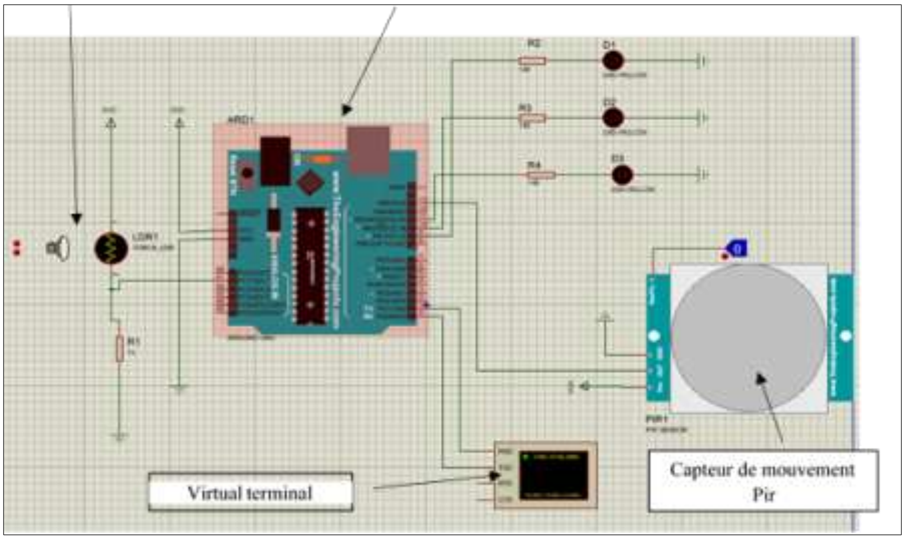


Figure 26 Diagram of an automatic lighting system with arduino

To vary the light intensity of an LED, we use PWM (Pulse Width Modulation) outputs. Arduino's 6 digital outputs (3, 5, 6, 9, 10, 11) can be used as PWM outputs. Figure 60 shows the operation of PWM signal. When a value of 255 is sent to the Pin of the LED, it continuously receives a value of 5V. The more the value received on the LED pin decreases, the more the voltage of the LED decreases.

In our case, we will connect an output LED to pins 9, 10 and 11. A light sensor (LDR) is plugged in as input to analog port A1 and a motion sensor Pir is plugged into a digital input (port 3). When the LDR receives a light less than or equal to 300 Lux, the LEDs light up with a minimum intensity (example 50% duty cycle, analogWrite(127)). In this case, the voltage received by the LEDs is not continuous here (Figure 60): alternating 5V and 0V at a regular frequency. At the same time, if motion is detected by the Pir sensor, the intensity of the LEDs increases to a maximum value (100% duty cycle, analogWrite(255)), 5V. The light intensity of the first LED increases in the first place, then a second after that of the second increases to maximum power, then a second after that of the third LED increases in power causing the decrease in the light intensity of the first LED. Otherwise, if the LDR receives a light intensity greater than 300 Lux, all LEDs go out even if the Pir picks up movement. In practice, the motion sensor is placed before the lamps for efficient lighting.

Otherwise, if the LDR receives a light intensity greater than 300 Lux, all LEDs go out even if the Pir picks up movement. In practice, the motion sensor is placed before the lamps for efficient lighting.

5. Conclusion

The concept of RCSF is always to minimize the energy consumed in the network regardless of the power mode used (battery, photovoltaic panels, SENELEC network). The use of these RCSF in public lighting is a beneficial contribution to reducing energy consumption. Beforehand, the lighting material is changed with LEDs. These are combined with connected objects. These two solutions coexist together and make "smart" street lighting possible. The term "smart" refers to the ability to control electricity consumption through the use of motion sensors for traffic control, light sensors to account for ambient light, and remote maintenance devices such as remote management software, lamp status monitoring (on or off) and remote connection to the lamp for low-cost maintenance. This can be summed up as lighting when necessary: passage of a pedestrian, a car, a cyclist or not. "Smart" street lighting reduces electrical energy consumption by 50 to 70% of initial consumption.

To have access to the data collected remotely, it is necessary that the data collected by the sensors can be transmitted via the Internet to the processing center for their operation and act accordingly according to the state of the lamps. In a LoWPAN network, an edge router handles this work. It is the intermediary between the RPL network and external networks.

The results obtained allowed us to deploy a 6LoWPAN RCSF network applied to smart street lighting

As a perspective, the physical experimentation of a public lighting system with microcontrollers compatible with 6LoWPAN and the large-scale implementation with more sensors (gas sensor, vibration sensor, temperature sensor, humidity sensor...) is beneficial for a city. Smart street lighting is the key to a smart city.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] http://www.syndicat-eclairage.com/wp-content/uploads/2017/03/SyndEclairage-dossier-SCM-n5-Eclairage-public.pdf

[2] Robert Benda, Thibault Fagiani, Paul Giovachini, Côme Pelée de Saint Maurice. “The city and the Internet of Things: Putting IoT at the service of the smart and sustainable city”. 2018. ⟨hal-01855829⟩

[3] Wafa Abdelghani, Florence Sèdes, Corinne Amel Zayani, Ikram Amous,” Detection of trust attacks in the Internet of Social Things”, 38th IT Congress of Organizations and Information and Decision Systems (INFORSID 2020), 2020, Dijon, France. pp.155-170. ffhal-03184015

[4] Yasmine Labiod “Security Mechanism for the Internet of Things”, 3rd cycle doctoral thesis Badji Mokhtar University- Annaba, 2022

[5] Sumeet Thombre, Raihan Ul Islam, Karl Andersson, and Mohammad Shahadat Hossain, “IP based Wireless Sensor Networks: Performance Analysis using Simulations and Experiments” Published 2016 Business, Computer Science J. Wirel. Mob. Ubiquitous Computing Networks. Dependable Appl.

[6] Fatima Zahra Saadaoui, Abderrahim Maizate, Mohammed Ouzzif, “State of the art on real-time protocols for the Internet of Things under the NDN network”. Conference on Connected Objects and Systems, Higher School of Technology of Casablanca (Morocco), University Institute of Technology of AixMarseille (France), Jun 2019, CASABLANCA, Morocco.

[7] ZACH SHELBY, CARSTEN BORMANN, “6LoWPAN: The Wireless Embedded Internet”, Online ISBN: 9780470686218, Wiley Telecom, DOI: 10.1002/9780470686218.refs

[8] Andrei Vasilateanu, Bogdan Bernovici, “Lightweight Smart Home Simulation System for Home Monitoring using Software Agents”, Procedia Computer Science 138 (2018) 153-160, www.sciencedirect.com

[9] Agus Kurniawan, “Practical Contiki-NG: Programming for Wireless Sensor Networks”, Apress Berkeley, CA, ISBN: 978-1-4842-3407-5 Published: 14 June 2018

[10] Cedric Chauvenet, “IPv6 support protocols for online powerline sensor networks”, Doctoral thesis University of Grenoble, 2013.