

Solid Waste Management – A smart IoT way

Pallavi K N, Ravikumar V, Ramesha Shettigar

Abstract: As the population is growing, the quantity of garbage is also multiplying proportionately. This big heap of accumulated garbage is making harm to the environment, spoils the location beauty and also creates health hazards in the area. The free open source simulation environment called Contiki OS helps in analyzing the performance parameters of Wireless Sensor Networks. The Cooja simulator in Contiki OS simulates the network scenario, provides the means, and tools to observe network behavior in different environments on different platforms. The results obtained by the simulator are found to be very near to actual deployment behavior. This paper explains the work done on analysis of RPL protocol in the applications of IoT. The sensor node reside inside the dust bin, it helps to obtain the status of the dustbin and routes information to reach those dustbins. To implement the shortest path between the dustbins, Travelling Salesman Problem (TSP) is used, which helps the truck driver to reach the dustbin in minimum time.

Index Terms: Contiki, Travelling salesman problem, RPL, Cooja, IoT

I. INTRODUCTION

As the population is growing, the quantity of garbage collected in rural as well as in urban is also multiplying proportionately. Similarly solid waste management has grown as a universal problem. It is required to take an appropriate action to get rid of this unmanageable solid waste [1]. There are three major sources where solid waste is produced. These sources are commercial, industrial and residential. The residential waste generated in that particular locality can be collected through home collection. The waste generated from hotels, hospitals, shopping complexes, movie theatres and other public buildings can be picked up from the establishment using various types of transportation fleets. The waste produced in construction sites, including by different industrial setups can be handled by various methods. The best practices of disposing of these wastes, like accumulation and effective handling of wastes, Internet of Things (IoT) concept is used, which mostly deals with sensing, actuating, data gathering, storing and processing by connecting physical and virtual devices to the Internet.

The new IPv6 protocol which is adopted has 128-bit addressability, likely to solve the addressability bottleneck faced by the IoT[2] phenomenon. IPv6 is emerging as a

communication standard for sensor networks of smart objects. In recent years, industry and academia have optimized IPv6 for low-power wireless systems based on the IEEE 802.15.4 standard, routing is a key part of the IPv6 stack that remains to be specified for such networks. RPL is the routing protocol for Low-power Lossy Networks (LLNs). In this protocol more than one root nodes in a network generates topology that drips downward to leaf nodes[3] is called as tree oriented protocol. Much flexibility is given in RPL, so that the constraints and metrics can be used in building a topology. In RPL there is a RPL border-router which will act as a gateway between the two networks. RPL border-router is used to obtain the status of the dustbins and routes information.

II. CONTIKI AND COOJA

A. Contiki operating system

Contiki is an open source operating system for networks of sensors. It is also useful for the networks [4], where wireless IoT devices are designed with an intention of consuming low power in memory systems. This is portable in nature. Various task can be carried out simultaneously and has inbuilt internet protocol (TCP/IP stack). It is designed such a way that it can be executed on various types of hardware platforms, especially which are suffering from memory limitations, runs by high power for processing and limitation of bandwidth for communication. There are three types of mechanisms by which network operate. They are uIP, TCP/IP and Rime stack [5]. uIP and TCP/IP stacks provides networks of IPv4 and IPv6. To make wireless system power efficient it requires communication primitives. This has to be given by Rime stack. IPv6 which is also known as 6LOWPAN in short is used to consume Low power wireless personal area networks. To simulate network[6], Contiki system introduced a simulator called Cooja.

B. Cooja Simulator

Contiki introduced a new simulator called cooja simulator. It is different from other simulator because it emulates the real hardware platforms. Cooja provides very useful method of simulating [7] and developing the Contiki code and system log before actually running on the target hardware. Developers debug their code and check the behavior regularly with new simulations for their systems. Cooja is a cross level simulator which means that it not only compiles native code but it also compiles Java codes. Usually Cooja compiles the C programs but it also compiles the Java programs by using Java Native Environment (JNE) to a compiled Contiki system. This makes simulation and deployment of the same code easy across different levels.

Manuscript published on 30 June 2019.

* Correspondence Author (s)

Pallavi K N, Computer Science and Engineering, NMAM Institute of Technology, Nitte, Udupi, India.

Dr. Ravikumar V, Computer Science and Engineering, Vidya Vardhaka College of Engineering, Mysuru, India.

Ramesha Shettigar, Computer Science and Engineering, NMAM Institute of Technology, Nitte, Udupi, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Also in Cooja the entire simulation can be saved and restore the simulation to where it was left off. Cooja does simulations in 3 levels namely, networking level, operating system and machine code instruction set level. At networking level the important factors are radio medium, radio devices, and sleep duty cycles of the sensor node. User can create or define new radio mediums. The applications written in Java can be ported to native with all functionalities[8]. A single node is simulated on only one level but the simulation of different nodes at different levels can be performed together. This gives Cooja the great advantage over other network simulators such as Network Simulator-2 (NS-2). Radio models can be used to test the nodes during simulation in specific environments. During the simulation, the range and interference can be changed. The radio absorb effect can also be seen in the future development of the simulator.

C. RPL border-router

RPL is known as IPv6 Routing protocol specially aiming to work for Low-power and Lossy Networks (LLNs). LLNs are class of network [9], includes routers and interconnections. LLNs routers in general work with constraints on processing power, energy and memory. RPL endow with mechanisms such as multipoint-to-point and point-to-multipoint. Traffic flows from devices contained by the LLNs to central control point is called multipoint-to-point. If the traffic flows from the central control point in the direction of the devices inside the LLNs is supported in point-to-multipoint.

RPL Border-routers are routers that originated at the boundary of the network shown in Fig 1. It is used to connect one network with other network. To obtain the RPL neighbors and routes information, set up of the IPv6 host for IP traffic bridging between host and 6LOWPAN [10] is necessary. TUNslip utility is provided in Contiki to connect RPL Network with an external IP network. The computer and the border-router are known as the gateway. Together they make it possible to transport packets from the outside world to the local 6LoWPAN-network, and backwards. The rpl-udp network is inside the 6LoWPAN network. When needs to specify the local 6LoWPAN prefix (aaaa::1/64), this will be used by all the nodes inside the network to identify their network. If it runs well, one must see the IPv6 address of the border-router, along with its corresponding local link IPv6 address.

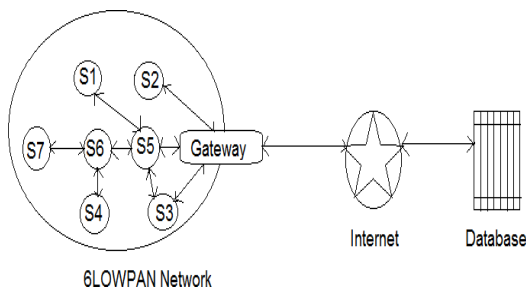


Fig 1: A typical 6LoWPAN topology

D. RPL border-router

RPL is known to be a proactive routing protocol and starts its function by searching routes immediately after

initialization of RPL[11] network. The nature of proactive routing protocol is, it provides the routes information, and it does not concern about whether it is needed for any node. This protocol helps to find and propagate the routes information once they start the network by periodically exchanging control messages. Nodes are sending local control messages to perform these two tasks: 1. To share local neighborhood information; 2. Message should send to the entire network for sharing the topology relevant data to all the nodes present in the network. Tree as topology Direct acyclic graph (DAG) formed by RPL. In RPL network every node has a selected parent which acts similar to a gateway for that node. If the routing table of a particular node is empty then it sends the packets either to its selected parent or to the common parent.

RPL belongs to distance vector protocol and it is divided into two phases based on the operations performed. They are routing upward and routing downward. Destination Oriented Directed Acyclic Graph [12] (DODAG) performs two operations: 1. Maintenance of topology information 2. It contains the path information from leaves to the root. The root is known as an LLN border router (LBR), the DODAG is considered to be grounded when it is connected to a non-LLN network (Private network).

E. Travelling salesman problem (TSP) algorithm

TSP is an NP-hard problem, it contains lists of cities and distances between the cities, here the main job is to find the shortest path among the cities and it should satisfy the condition that it visit each city exactly once and return to the starting city[13]. TSP is categorized into two groups based on the cost matrix structure:

Symmetric: Consider two cities A and B; the distance from A to B is same as the distance from B to A. As a result this condition forms an undirected graph. The TSP is symmetric if $C_{ij} = C_{ji}$ for all i, j . There is $(n-1)! / 2$ feasible solutions, where n is number of cities.

Asymmetric: Consider two cities A and B, the distance from A to B is different from the distance from B to A, sometimes routes may not be existed in both ways. There is $(n-1)!$ feasible solutions exist, where n represents number of cities.

III. PROPOSED WORK

Now solid waste management is a top priority in the big cities of the world. As the population is growing, the quantity of garbage is also multiplying proportionately. This big heap of accumulated garbage is making harm to the environment, spoils the location beauty and also creates health hazards in the area. The free open source simulation environment called Contiki OS helps in analyzing the performance parameters of Wireless Sensor Networks. The Cooja simulator in Contiki OS simulates the network scenario, provides the means, and tools to observe network behavior in different environments on different platforms. The results obtained by the simulator are found to be very near to actual deployment behavior.



IV. RESULTS AND DISCUSSION

In this paper setting up of the RPL border-router, display of status of the dustbins and routes information, sending and receiving notification at particular mote and execution results are discussed. This portion of the paper explains the mote to know the status of the dustbin for the scenario1.

Parameters	Values
Channel type	Wireless channel
Radio medium	Unit Disk Graph Medium (UDGM) : Distance Loss
Antenna	Omni direction
Antenna range	50m
MAC type	IEEE 802.15.4
Medium access techniques	CSMA/CA
Number of nodes	7

Table 1: Node configuration for scenario 1

Table 1 gives the details of node configuration for scenario 1. The nodes implement multi-hop technique to forward data if the server node is far off. The multi-hop technique helps in sending the data with low power and thus reduces the power consumption.

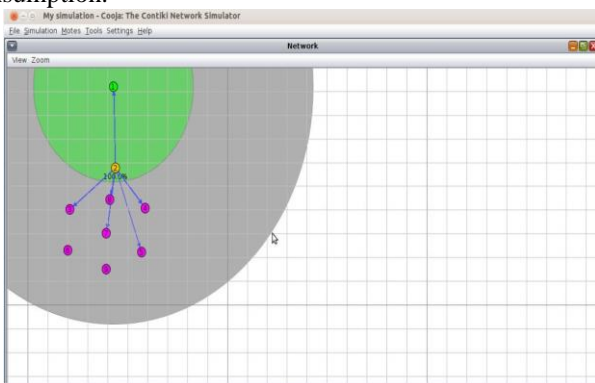


Fig 2: Network Window-Node arrangement-Scenario1

Based on the scenario1 the node formation of RPL border-router and rpl-udp is as shown in the Fig 2. It indicates one border-router node, one udp-server node and seven udp-client nodes. The green circle represents the transmission range of the selected node. Two nodes are inside the green circle it means those nodes within the transmission range and the other nodes within gray circle represent the interference range. In this node 1 (green color) representing border-router and act as a gateway, node2 (yellow color) is udp-server and act as a cluster head, the remaining nodes from 3 to 9 (pink color) are udp-client and acts as a dustbin.

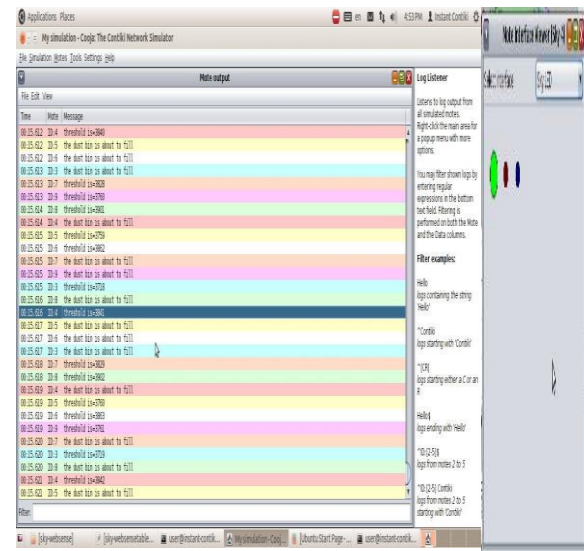


Fig 3: The result of Mote output and Mote interface viewer (node4)

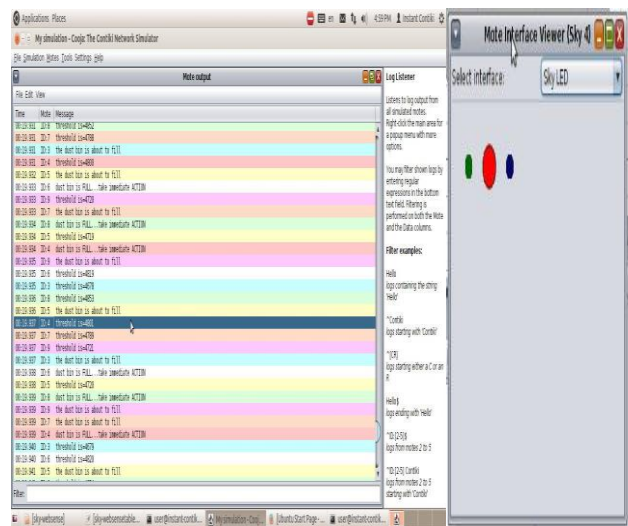


Fig 4: The result of Mote output and Mote interface viewer (node4)

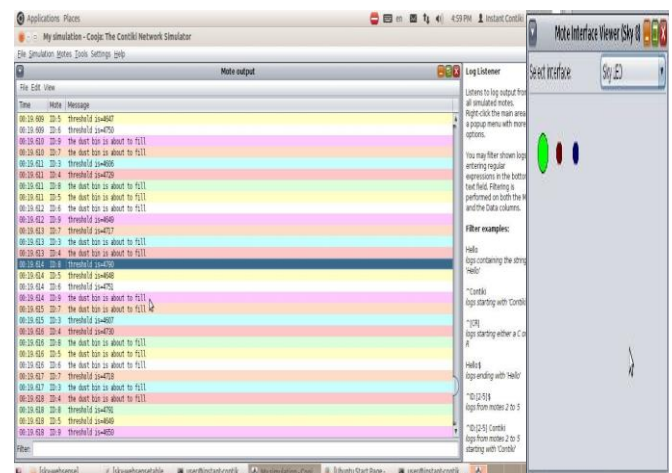


Fig 5: The result of Mote output and Mote interface viewer (node8)

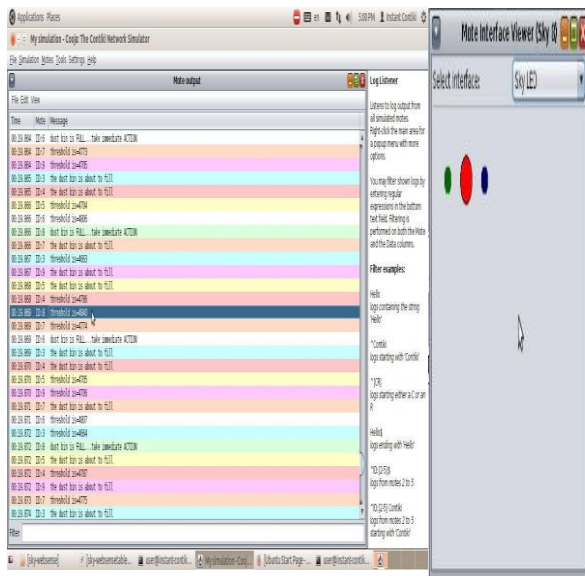


Fig 6: The result of Mote output and Mote interface viewer (node8)

When the dustbin threshold level reaches to the mark of 2500 LVL and above, the green LED will turn ON automatically and by sensing this LED the dustbin sensor node send the notification to the border-router through cluster head. The notification contains the “dustbin is about to fill” message. When the dustbin threshold level reaches to 4800 and above 4800, the red LED will turn ON and by sensing this LED the dustbin sensor node send the notification to the border router through cluster head. The notification contains the “dustbin is full...take immediate action” message. The outcome is as shown in the Fig 3, Fig 4, Fig 5 and Fig 6.

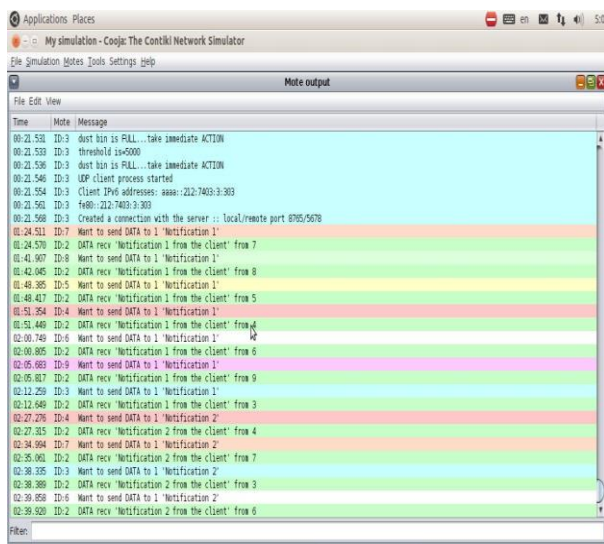


Fig 7: Mote output window-Scenario 1

When udp-client wants to send the message to the border-router, it will receive by the cluster head then the cluster head sends that message to the border-router. The outcome is as shown in Fig 7.

Applications Places

Contiki Web Sense - Mozilla Firefox

Ubuntu Start Page x Contiki Web Sense x +

10.0.0.1:212.7404.4.404

Search

Status of the Dustbin summary table

Dustbin ID	Threshold value	Message	LED color
4	1668	No action required	BLUE
5	2957	Dustbin is about to fill	GREEN
6	4815	Dustbin is Full...take immediate action	RED
7	3098	Dustbin is about to fill	GREEN
8	4790	Dustbin is about to fill	GREEN
9	1801	No action required	BLUE

Fig 8: Status of the dustbins- Scenario1

Fig 8 shows the status of the dustbins located in a particular area. It displays the status of the dustbin including information such as threshold value of the dustbin, message received in border router, LED color, and dustbin ID.

The second scenario is given in the Table 2:

Parameters	Values
Channel type	Wireless channel
Radio medium	Unit Disk Graph Medium (UDGM) : Distance Loss
Antenna	Omni direction
Antenna range	50m
MAC type	IEEE 802.15.4
Medium access techniques	CSMA/CA
Number of nodes	10

Table 2: Node configuration for scenario 2

Table 2 gives the details of various parameters and their values considered in scenario 2.

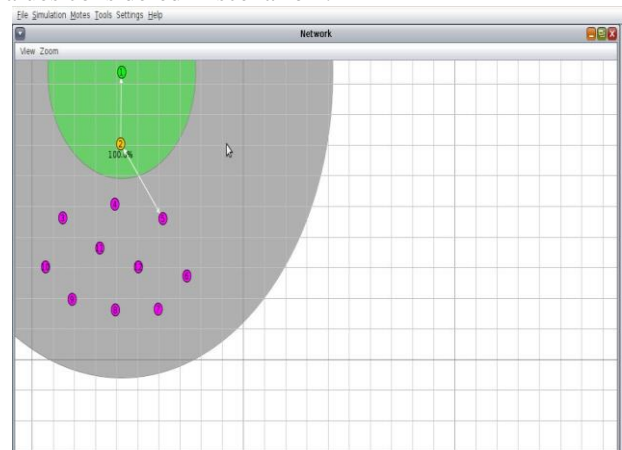


Fig 9: Network Window-Node arrangement-Scenario 2

Fig 9 shows the node formation of RPL border-router and rpl-udp. It indicates one border-router node, one udp-server nodes and ten udp-client nodes.

Here the green circle represents the transmission range of the selected node, in the above Fig 9 two nodes are inside the green circle it means those nodes within the transmission range and the other nodes within gray circle represent the interference range. In this screen shot node 1 (green color) representing border-router and act as a gateway, node 2 (yellow color) is udp-server and act as a cluster head, the remaining nodes from 3 to 12 (pink color) are udp-client and acts as a dustbin.

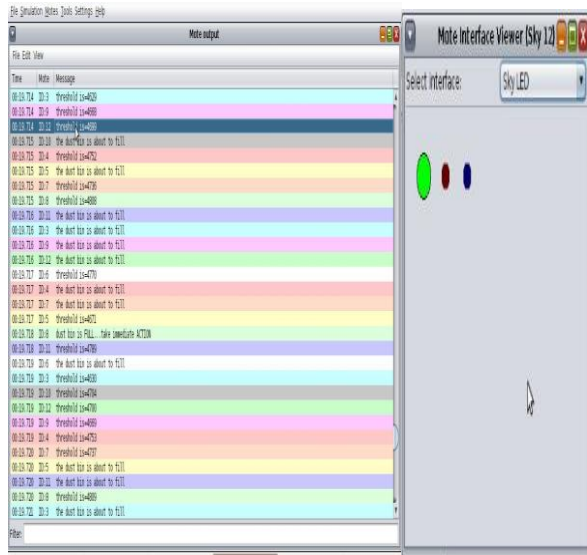


Fig 10: The result of Mote output and Mote interface viewer (node12)

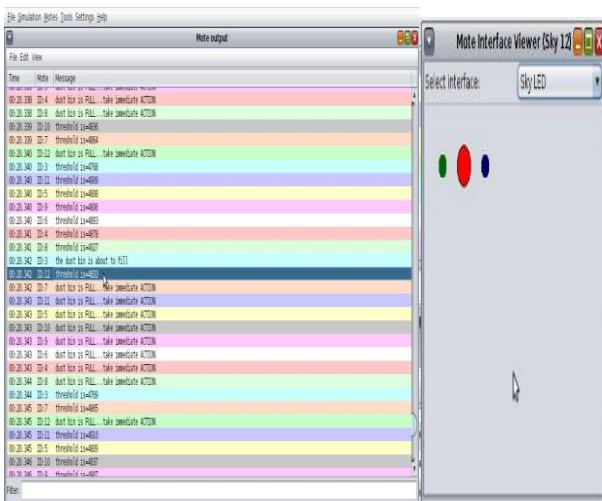


Fig 11: The result of Mote output and Mote interface viewer (node12)

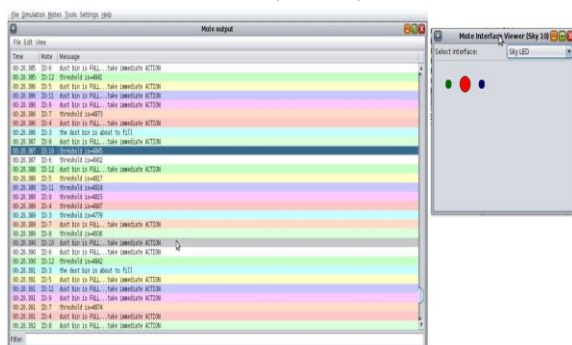


Fig 12: The result of Mote output and Mote interface viewer (node10)

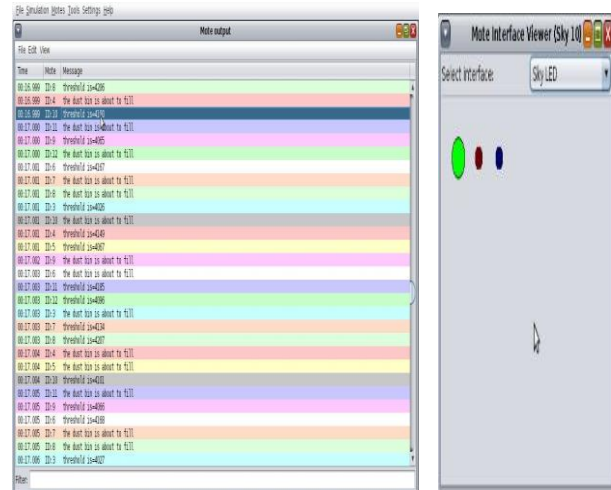


Fig 13: The result of Mote output and Mote interface viewer (node10)

When the dustbin threshold level reaches to 2500 and above 2500, the green LED is turn ON and by sensing this LED the dustbin sensor node send the notification to the border-router through cluster head. The notification contains the “dustbin is about to fill” message. When the dustbin threshold level reaches to 4800 and above 4800, the red LED is turn ON and by sensing this LED the dustbin sensor node send the notification to the border router through cluster head. The notification contains the “dustbin is full... take immediate action” message. The outcome is as shown in the Fig 10, Fig 11, Fig 12 and Fig 13.

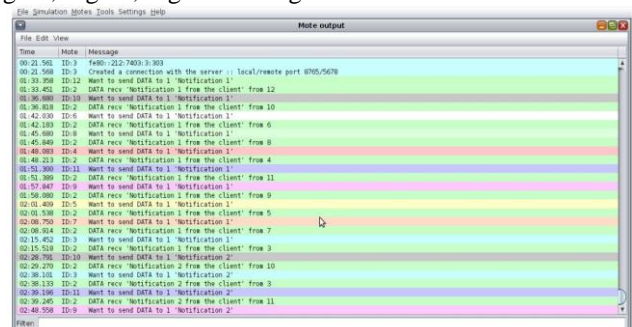
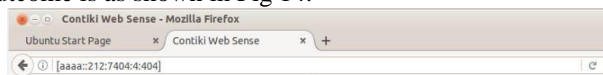


Fig 14: Mote output window - Scenario 2

When udp-client wants to send the message to the border-router, it will receive by the cluster head then the cluster head sends that message to the border-router. The outcome is as shown in Fig 14.



Status of the Dustbin summary table

Dustbin ID	Threshold value	Message	LED color
4	4844	Dustbin is Full..take immediate action	RED
5	1717	No action required	BLUE
6	3049	Dustbin is about to fill	GREEN
7	1719	No action required	BLUE
8	1889	No action required	BLUE
9	1642	No action required	BLUE
10	4825	Dustbin is Full...take immediate action	RED
11	4878	Dustbin is Full...take immediate action	RED
12	3037	Dustbin is about to fill	GREEN

Fig 15: Status of the dust bin- Scenario 2

Fig 15 shows the status of the dustbins located in a particular area. It displays the status of the dustbin including information such as threshold value of the dustbin, message received in the border router, LED color, and dustbin ID.

A. Obtaining the shortest path – Scenario 1

This part explains how to obtain the shortest path using travelling sales man problem for the scenario1



Fig 16: a) Status of the Dustbin - Scenario1

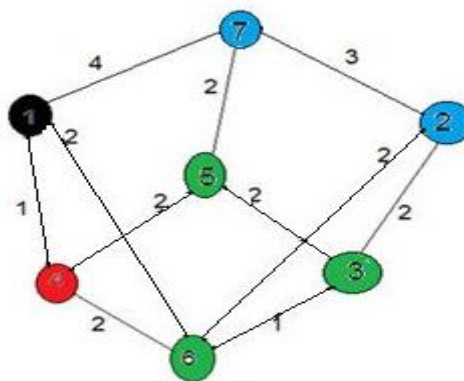


Fig.16: b) weighted graph - Scenario 1

When the IPv6 address of the any dustbin is entered, the output displayed is shown in Fig 16: a). Here the statuses of all six dustbins are displayed. The information includes threshold level of the dustbin, message received in the border router, LED color, and dustbin ID. In Fig 16: b) shows the weighted graph where each node represents the dustbin and cost is nothing but the link cost that is distance between the dustbins in particular area. Here the node1 with black color denotes dumping area, the truck driver will starts and end his journey at the node1 and nodes with blue color denotes empty dustbins, nodes with green color denotes half filled dustbins and nodes with red color denotes fully filled dustbins.

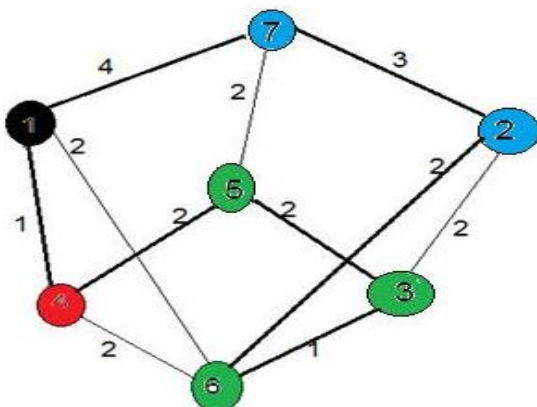


Fig 17: Shortest path when considering all the dustbins- Scenario1

In Fig 17 the dark lines represent the shortest path by considering all the dustbins and the truck driver should visit all the dustbins only once and return back to the starting point that is dumping yard.

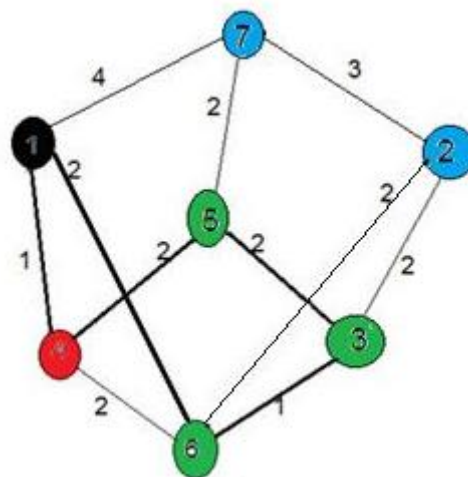


Fig 18: Shortest path when considering fully filled and half filled Dustbins – Scenario 1

In Fig 18 the dark lines represent the shortest path when considering only the half filled and fully filled dustbins and the truck driver should visit only those selected dustbins exactly once and return back to the starting point that is dumping area.



Fig 19: Shortest paths – Scenario 1

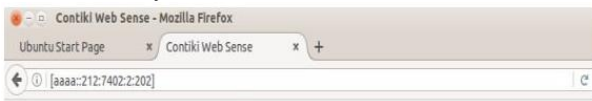
When the IPv6 address of the any dustbin is entered the output is displayed as shown in Fig 19. Here the route information in both cases can be tracked. In first case, all the dustbins are considered, path and cost information is computed. In second case, only half filled and fully filled dustbins are considered, cost and path information is computed.

B. Obtaining the shortest path – Scenario 2

This part explains how to obtain the shortest path using travelling sales man problem for the scenario2.

When the IPv6 address of the any dustbin is entered, the output displayed is shown in Fig 20: a). Here the statuses of all nine dustbins are displayed. The information includes threshold level of the dustbin, message received in the border router, LED color, and dustbin ID. In Fig 20: b) shows the weighted graph where each node represents the dustbin and cost is nothing but the link cost that is distance between the dustbins in particular area.

Here the node1 with black color denotes dumping area, the truck driver will starts and end his journey at the node1 and nodes with blue color denotes empty dustbins, nodes with green color denotes half filled dustbins and nodes with red color denotes fully filled dustbins.



Status of the Dustbin summary table

Dustbin ID	Threshold value	Message	LED color
2	1668	No action required	BLUE
3	2957	Dustbin is about to fill	GREEN
4	4815	Dustbin is Full..take immediate action	RED
5	3098	Dustbin is about to fill	GREEN
6	4790	Dustbin is about to fill	GREEN
7	1801	No action required	BLUE

Fig 20: a) Status of the Dustbin - Scenario2

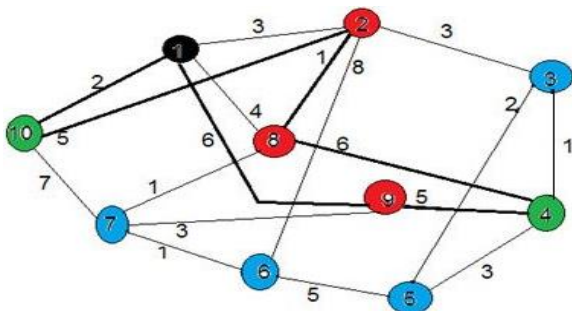


Fig 20: b) weighted graph – Scenario 2

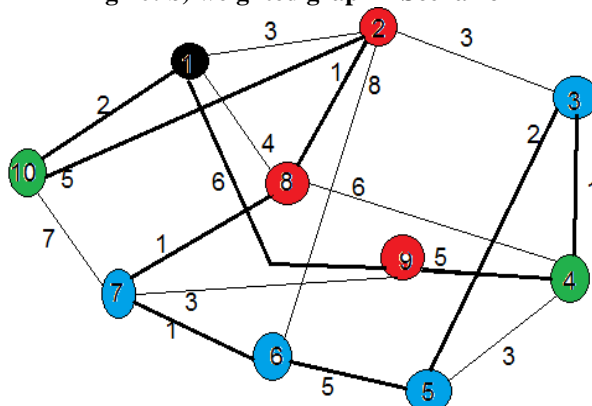


Fig 21: Shortest path when considering all the dustbins – Scenario 2

In Fig 21 the dark lines represent the shortest path by considering all the dustbins and the truck driver should visit all the dustbins only once and return back to the starting point that is dumping yard.

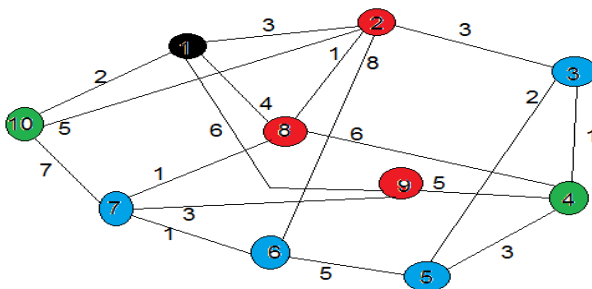
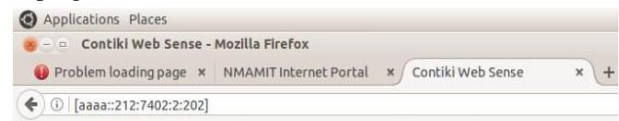


Fig 22: Shortest path when considering fully filled and half filled Dustbins – Scenario 2

In Fig 22 the dark lines represent the shortest path when considering only the half filled and fully filled dustbins and the truck driver should visit only those selected dustbins exactly once and return back to the starting point that is dumping area.



Status of the Dustbin summary table

Dustbin ID	Threshold value	Message	LED color
2	4800	Dustbin is Full..take immediate action	RED
3	1400	No action required	BLUE
4	2500	Dustbin is about to fill	GREEN
5	910	No action required	BLUE
6	1050	No action required	BLUE
7	950	No action required	BLUE
8	4800	Dustbin is Full..take immediate action	RED
9	4800	Dustbin is Full..take immediate action	RED
10	2500	Dustbin is about to fill	GREEN

Fig 23: Shortest paths – Scenario 2

When the IPv6 address of the any dustbin is entered in the output is displayed as shown in Fig 23. Here the route information of both cases can be tracked. In first case, all the dustbins are considered, path and cost information is computed. In second case, only half filled and fully filled dustbins are considered, cost and path information is computed.

V. CONCLUSION AND FUTURE WORK

The resulting Cooja simulator works well for the uIP protocol stack such as IPv6, RPL. RPL separates packet processing and forwarding from the routing optimization objectives. Collection of data like packet transformation, power consumption, adding and removing nodes at any time is possible in Cooja simulator. It allows new users for quick and ease start up a simulation and it is most helpful in development and test phases. It also supports big scale behavior protocols and algorithms.

Using RPL border-router it is observed that, we get the status of the dustbin and routes information by specifying the IPv6 address on the web browser. By using LED features present in the Contiki, When LED glows, the sensor node send the notification to the border-router via cluster head. Then it alert the truck driver by sending message for the collection of garbage immediately. As a result we can avoid the overflowing of garbage in dustbin. By using TSP shortest path algorithm, it determined the shortest path among the dustbins placed in specified area which helps the truck driver to reach the dustbin in minimum time.

In the future work of this project, IOT based Wireless Sensor Networks can be simulated to separate the waste into different categories such as dry (plastic, paper) and wet (vegetables). The Contiki OS is open source and the codes can be further modified and improved to get a better performance.

REFERENCES

1. Fallavi, K.N.; Kumar, V.R.; Chaithra, B.M. Smart waste management using Internet of Things: A survey. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017; pp. 60–64.
2. Aggarwal, C. C., Ashish, N., & Sheth, A. (2013). The Internet of Things: A survey from the data-centric perspective. In C. C. Aggarwal (Ed.), *Managing and mining sensor data* (pp. 383–428). US: Springer.
3. T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," March 2011.
4. Velinov, A.M.A.: Running and testing applications for Contiki OS using Cooja simulator. In: International Conference on Information Technology and Development of Education – ITRO (2016)
5. A. Dunkels, "Poster Abstract: Rime — A Lightweight Layered Communication Stack for Sensor Networks," in Proceedings of EWSN, Delft, The Netherlands, 2007.
6. The Official Contiki OS Blog [Online]. Available: <http://contiki-os.blogspot.mk/>. (Access Date: 29.03.2016)
7. K. Roussel, Y.Q. Song, O. Zandra, "Using Cooja for WSN Simulations: Some New Uses and Limits", Kay Roemer. EWSN 2016 - NextMote workshop, Feb 2016, Graz, Austria
8. Junction Publishing, Proceedings of the 2016 International Conference on Embedded Wireless Systems and Network - EWSN '16 - NextMote workshop, pp.319-324, 2016.
9. Osterlind, F; Dunkels, A; Eriksson, J; Finne, N; Voigt, T. Cross Level Sensor Network Simulation with Cooja. Proceedings of the 31st IEEE Conference on Local Computer Networks (LCN), Tampa, FL, USA, 14–16 November
10. Umamaheswari, S.; Negi, A. Internet of Things and RPL routing protocol: A study and evaluation. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 5–7 January 2017.
11. S. Raza, S. Duquenois, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing Communication in 6LoWPAN with Compressed IPsec," in IEEE International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS2011), Barcelona, Spain, Jun. 2011.
12. Hazrat Ali, "A Performance Evaluation of RPL in Contiki "Master Thesis, School of Computing, Blekinge Institute of Technology.
13. Abdel Hakeem, S.A.; Hady, A.A.; Kim, H. RPL Routing Protocol Performance in Smart Grid Applications Based Wireless Sensors: Experimental and Simulated Analysis. *Electronics* **2019**, *8*, 186.
14. Varshika Dwivedi, Taruna Chauhan, Sanu Saxena and Princie Agrawal, "Travelling Salesman Problem using Genetic Algorithm", National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI) 2012.

AUTHORS PROFILE



Mrs. Pallavi K N is a research scholar at department of Computer Science and Engineering, Vidya Vardhaka college of Engineering, Mysuru. She is working as a Assistant Professor at NMAM Institute of Technology, Nitte. She is having 12 years of experience in teaching. Her areas of interests are Internet of Things, Cloud Computing, Fog computing and network security.



National/International journals/Conferences

Dr. Ravikumar V professor and head in the department of Computer Science and Engineering, Vidya Vardhaka College of Engineering, Mysuru. He is having 17 years of teaching and 4 years of research experience. His areas of interest are Bigdata, Cloud Computing, Internet of Things, NLP and Artificial Intelligence. He published more than 19 papers in



Mr. Ramesha Shettigar is working as a Assistant Professor at NMAM Institute of Technology, Nitte. He is having 10 years of experience in teaching. His areas of interests are Internet of Things, Cloud Computing.