Geographic Location based Dynamic and Opportunistic RPL for Distributed Networks

Manali Chakraborty[1], Alvise Spano[1], Agostino Cortesi[1]

Università Ca' Foscari, Via Torino, 153, 30172 Venezia VE
manali.chakraborty@unive.it, alvise.spano@unive.it, cortesi@unive.it

Abstract. RPL or Routing Protocol for Low power and Lossy Networks (LLNs) is considered the most suited routing technology for IPV6. However, with the rapid advancements in networking and the paradigm shift towards IoT, RPL is facing some performance issues due to scalability, resource constraints and mobility. We propose a Geographic location-based Dynamic Opportunistic routing protocol (GDO-RPL) for point-to-point communication, whose algorithm has been extensively simulated within the Contiki Cooja environment. As a testbed for the proposed solution, we evaluated the scenario of an innovative 3D printing system that uses IIoT (Industrial Internet of Things) technologies, such as sensors installed within artifacts collecting data during the lifetime of the object in the environmental context for which it is intended.

Keywords: IoT; RPL; Additive Manufacturing; Opportunistic Routing; Distributed networks.

1 Introduction

Along with the rapid advancement of Internet of Things (IoT), IPv6 becomes an obvious choice for implementing distributed networks. We consider in particular for underlying routing protocol, RPL, which is a distance-vector routing protocol and based on IPv6 for Low-power and Lossy Networks (LLNs). It was designed considering the requirements specified in RFC 5826 [2], RFC 5673 [3], RFC 5548 [4] and RFC 5867 [5].

This LLNs have the following specific characteristics [1]:

- it consists of a group of constrained nodes, which have limited processing and storage capabilities;
- nodes are connected via lossy links, typically supporting only low data rates, that are usually unstable with relatively low packet delivery rates;
- the traffic patterns are not simply point-to-point, but in many cases point-to-multipoint or multipoint-to-point;
- furthermore, such networks may potentially comprise up to thousands of nodes.

1.1 Motivation

As outlined by some recent works on RPL discussed in section II, even though RPL is still recognised as the most suitable routing protocol for IoT and IPv6 at the moment, it still has some flaws, among which:

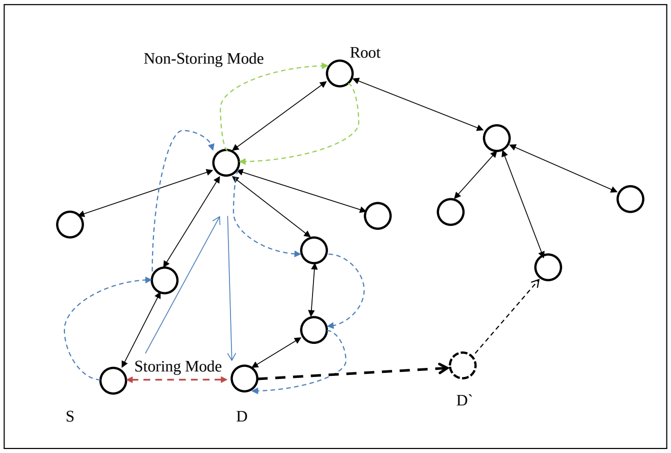


Fig. 1. Overhead in RPL route selection process.

1. In P2P communication, storing mode, a data packet travel upwards until a common ancestor and then return downwards to the destination, and in non-storing mode, it goes directly to the root and then travel downwards. This strategy creates congestion close to the root and greatly increases overhead and latency. In figure 1, we can see the overhead for both storing and non storing mode for P2P communication between S and D.
2. Another major concern for RPL is mobility. A mobile node can join any existing DODAG, at any point. However, in point-to-multipoint communication, as the mobile node moves from a parent to another, information disseminated via DAOs may rapidly become obsolete.

   Suppose, node D in figure 1 moves to D', then the present common ancestor cannot find a path to D' and there will be packet drop.
3. Scalability is another issue in RPL. In non-storing mode, the limiting factor is the size of the packet header containing the source routing information; this can include up to 8 IPv6 addresses (64 with a compressed header), but the longer the header the higher the overhead and the route repair latency. In storing mode, instead, the limiting factor is the memory available to store neighbor and routing tables. The nodes close to the root must store routing state for almost the entire DODAG, which can be challenging for resource-constrained devices.

Now, RPL is the tailor made solution keeping in mind the requirements of IPv6 based networks. However, these requirements were published 7 years ago [1], a relatively long time in the fast-paced world of networked computing in general and IoT in particular. Hence, with the rapid advancement in networking domain and the paradigm shift towards the IoT, it is high time to address these above mentioned issues.

1.2 Contribution

In this paper, we propose a geographic location based, dynamic and opportunistic RPL. This extension of RPL, generally aims to reduce the memory overhead and mobility issues in P2P communication. Instead of using the Root of the DAG or a common ancestor for both sender and receiver node, two nodes can communicate directly, or via some leaf nodes of the DAG. The selection of the routes are based on the geographic location, and the degree of the routes are chosen dynamically, such that they always ensure minimum connectivity between two nodes. Besides, the selection of routes follow the opportunistic routing mechanism [6], where a set of candidate node selects the most deserving node to transmit a data packet. The proposed algorithm is extensively tested using Contiki-Cooja.

1.3 Application Scenario

Additive Manufacturing (AM) or 3D printing is slowly gaining popularity as one of the most interesting and innovative domain for research purposes. Generally AM is used to develop 3D models, though the associated cyber physical system (CPS) can be used for several applications [7]. In this paper we have considered a scenario where IoT based network is used to collect information using sensors, placed on 3D artifacts designed by the 3D printers. These sensors can communicate between themselves and with a base station. The goal is to improve the production process by analyzing such data in order to make the overall communication system more efficient, robust and scalable.

1.4 Structure of the paper

The rest of the paper is structured as follows: Section II discusses some relevant works in this domain, while the detailed methodology of our algorithm is described in section III. Section IV presents the results and finally, section V concludes the paper.

2 State of the Art

In RPL, three types of communication are supported;

- P2M or point to multipoint, from root node to lower level nodes in the DODAg. This communication goes downwards.

- M2P or multipoint to point, from nodes to root.
- P2P or point to point, between two distinct nodes in a network.

In order to achieve this three types of communication, RPL has two different modes of operation;

- Storing mode: In this mode, each node keeps a routing table containing mappings between all destinations reachable via its sub-DODAG and their respective next-hop nodes, learned while receiving DAOs.
- Non-storing mode: In this mode, the root is the only network node maintaining routing information.

Authors of [12], try to improve the performance of RPL under storing mode by improving the memory consumption of each node. They restrict the number of routing table entries for every storing node, by a predefined number N. If a storing node reaches the value N, it then create a new storing node among its children, and the process continues, until all the storing node are exhausted, and the mechanism turns into typical storing mode. This method tries to lower the burden of storing nodes, specially those are closer to roots, however, with increasing scalability and mobility, the routing tables at every storing node fail to capture the exact topology of surroundings.

In storing mode, if a node send a DAO and its parent can't store its information for memory issues, then the parent sends a DAO-NACK, to notify the node. The authors of [15], propose a multicast mechanism, where a node or a set of nodes, which can't store the information of their sub DODAG in the routing table anymore, joins a multicast group and declares themselves as junction node. When the root wants to communicate with a leaf node, for which it has no path, the root then multicasts the message to the multicast group. The members of that group then look for the destination in their routing tables and forward the packet accordingly. This method improves the performance of RPL in storing mode by changing the root's behaviour.

However, it still suffers from scalability and mobility issues [12]. Besides, as the size of the network grows, more messages are sent as broadcast message to every junction node, and inturn the junction nodes have to look up their routing table for the destination. Hence the cost of packet delivery increases.

Another modification over RPL has been proposed in [10] . This paper tries to improve the shortcomings of [12] and [15] by using a hybrid mode of operation for downward routing, i.e., for P2M and P2P routing. They distributes the routing information over root node and routing nodes. The root node stores the information of all the destination nodes, whereas, the router nodes store a routing table containing information about other routeing nodes. The P2P communication is performed using non-storing mode, (leaf-root-leaf), whereas the router to router communication is performed using storing mode. Authors claim that this method overcomes the disadvantages of both storing and non storing modes by showing lesser routing table entries than storing mode and decreasing the end-to-end delay than non-storing mode. However, the proposed method experiences performance degradation with mobility in the network.

In [16], an extension of RPL is proposed, that improves the performance of non-storing mode. It proposes that, when a node wants to send a packet to another node, then instead of sending it upwards to the root, it broadcasts a Route Request message (like AODV) and tries to build a DODAG originated at itself. In this way using a reactive path discovery method, the proposed work improves the traffic density near the root and ensures a better PDR by decreasing delay and hop count.

RPL is no doubt the best suitable choice for routing in IPv6 based networks with constraied nodes. However, with the advancement in IoT, RPL is facing some performance issues, due to scalability, resource constraint of nodes and scalability. The works discussed above try to propose extensions based on RPL, to overcome this problems. However, handling mobility in RPL is still an open problem.

2.1 Aim of the paper

The primary objective in this paper is to design an extension of RPL which can perform efficiently in a scalable and mobile network. Now, RPL is a toplogy based routing or table based routing, where various nodes in the network (storing mode) or only the root (non-storing mode) stores the topology of the network and use this information to maintain a stable communication system in the network. Now this topology based routing methods stores the topology of the network, which is not very helpful for rapidly changing networks. On the other hand, geographic location based routing protocols are proved to be an improvement for this types of networks, as only the location information is needed for routing.

Location-based routing based methods are depending on the fact that every node knows its location, as well as the destination node's location. It has low communication overhead because advertisements of routing tables, like in traditional routing protocols, are not needed. Therefore, Location-based routing conserves both energy and bandwidth since route request and state propagation are not required after one-hop distance. Location-based routing uses the location information for nodes to provide higher efficiency and scalability. However, the location information always have some degree of inaccuracy in it. Specially, if the network supports mobility, then its very difficult to keep track with the nodes' locations. Another major drawback of geographic location based routing is fault tolerance, which results in poor throughput and PDR.

Hence we tried to merge this two types of routing in such a way that each of them can make another one stronger and results in a better routing for IPv6.

3 Proposed Solution

In this paper, we propose a geographic location based, dynamic routing protocol for point to point communication.

We assume that, every node in this system stores the information of its one hop neighbours as depicted in figure 2(c). However, broadcasting the message to this entire neighbour list can still incur a great overhead on both computation and memory. Besides, its completely absurd to involve a node in the routing process if its on the exactly opposite direction of the destination node. Thus, we proposed a geographic location based routing, where the nodes will communicate with their one hop neighbours which are on the way to the destination node.

3.1 Information stored at each node

Every node N should store the following informations:

1. Its own geographic address. Let (x_i, y_i) denotes the address of node N_i.
2. Address of its one hop neighbor nodes.
3. Address of the destination node.

Let us consider a node S. If we consider the one hop neighbors of S as,

$$S_N = \{S'_1, S'_2, \ldots, S'_M\}$$

where, M is the total number of one hop neighbours of S.

Given α, S_N^α is the set of neibouring nodes of S, which falls under the angle α. In figure 2 the neighbour nodes of S for 360°, 180°and 90°have been shown.

Every node will select the degree of angel based on the total number of neighbouring nodes. Our main purpose of this work is to maintain a equilibrium between performance and resources. Hence, we provide dynamic node selection for next hop transmission, considering a certain level of performance assurance. If we have selected 180°as an angel, then the probability of a single node in the neighbourhood of being on that 180°angle is 1/2. Similarly, the probability of a single node to remain in 90°angle is 1/4.

Hence, in order to assure atleast one path towards the destination, the collective probability should be greater than 1.

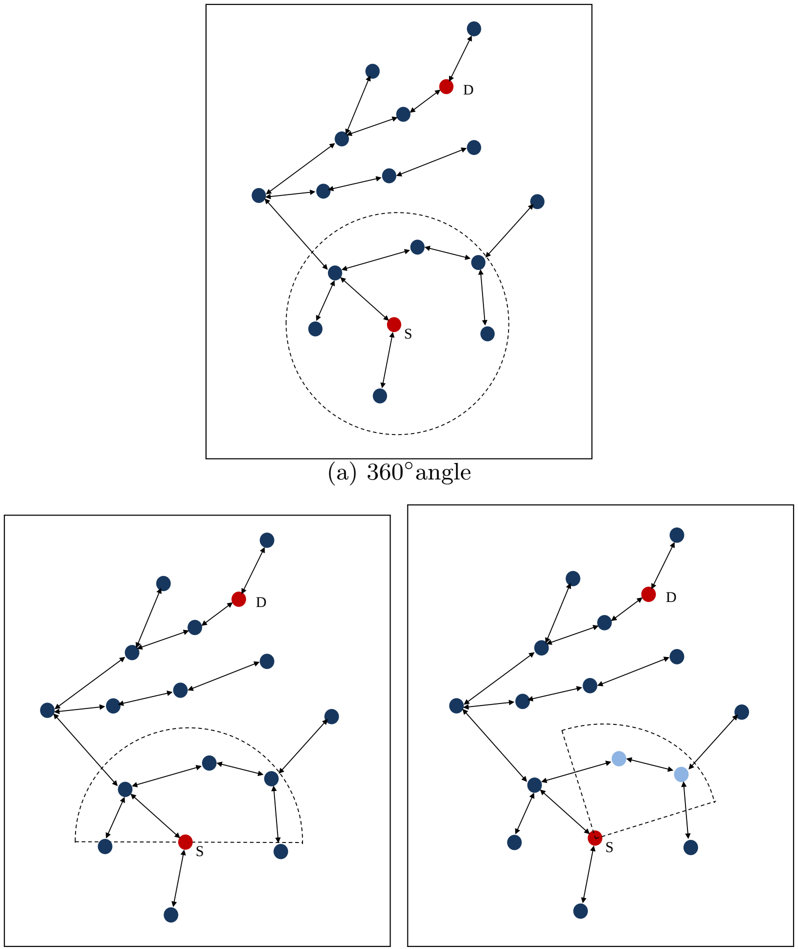
Thus,

$$\mid S_N^\alpha \mid \leq \tfrac{2\pi}{\alpha}$$

3.2 Neighbour Discovery

Every node, when it wants to join a network, broadcasts DIO message to find its DODAG instance and preferred parent. Each DIO message contains the predefined routing metric of the network, such as path reliability, hop count, bandwidth, latency, etc. and an objective function which determines the preferred parent towards the root for every node. Since a node broadcasts this DIO message, nodes in its one-hop distance can listen to this message, and update its neighbour tables accordingly.

Now, this DIO messages are rebroadcasted by each node according to an adaptive technique, the Trickle algorithm [6], which strikes a tradeoff between reactivity to topology changes and energy efficiency. Trickle ensures that DIOs are advertised aggressively when the network is unstable, and instead rebroadcast at an increasingly slow pace while the network is stable. This ensures that every node remains updated with its surrounding topology.



(a) 360°angle

(b) 180°angle

(c) 90°angle

Fig. 2. Neighbor selection with different angles

3.3 Route Establishment

Suppose a node S_0 wants to send a packet to a destination node D: the algorithm 1 runs on each node of the network, implementing a routing mechanism: nodes are dormant and awake when a unicast message is received, triggering the execution of the algorithm; the only exception is the starting node S_0 initiating the whole iteration. It is capable of routing a message M coming from a neighbor source node S to a target node among its neighbors in a recursive way; base case is when the destination node D directly belongs to the neighborhood. Please note that the information of the initial node S_0 is not forwarded to subsequent nodes, and $S \equiv S_0$ only for the second iteration.

Moreover, the algorithm assumes a discovery process has already taken place: every node must know its own address, as well as its one-hop neighbour addresses and their geographical coordinates. In the pseudo-code some function calls appear, whose semantics are straightforward:

GetMyNeighbors returns the neighbors of the self node, which have previously been discovered.

Unicast unicast the message argument to the given node.

InitAngle calculates the starting, ending, increment and direction angles in function of the source and self nodes. Parameters α_1, α_2 and δ could be constants such as 10°, 180° and 10° respectively, or even be the result of a finer computation. ϕ is calculated as the direction from which source node S originates with respect to the self node: assume (S_x, S_y) and $(\texttt{self}_x, \texttt{self}_y)$ are the coordinates of S and self respectively, then:

$$\phi = \arctan 2(\texttt{self}_y - S_y, \texttt{self}_x - S_x)$$

RestrictNeighbors calculates the subset of nodes laying within the angular range given as pair of arguments.

Length calculates the length of a list of nodes.

Farthest calculates the farthest node among those passed as argument, comparing each one's distance from self.

Algorithm 1: RouteMessage(M, S, D)

```
input: the message $M$
input: source node $S$
input: destination node $D$
let self be the current node;
nodes ← GetMyNeighbors();
if $D \in$ nodes then
    Unicast($M$, $D$);
else
    $(\alpha_1, \alpha_2, \delta, \phi)$ ← InitAngle(self, $S$);
    for $\alpha = \alpha_1$ to $\alpha_2$ step $\delta$ do
        restricted ← RestrictNeighbors(nodes, $\phi - \frac{\alpha}{2}, \phi + \frac{\alpha}{2}$);
        if Length(restricted) $\geq \frac{2\pi}{\alpha}$ then
            nodes ← restricted;
            break;
        end
    end
    target ← Farthest(nodes);
    Unicast($M$, target);
end
```

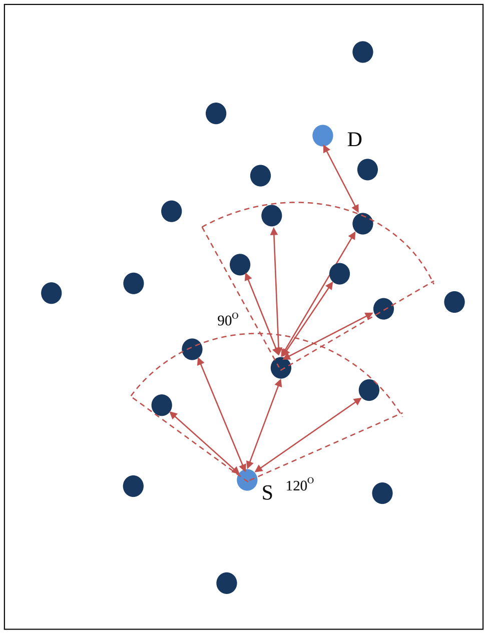


Fig. 3. Route selection process.

4 Results

We simulate our proposed algorithm using Contiki-Cooja [13, 14]. The simulation settings are described in table 1. Figure 4 shows the variation of total time needed to transmit a packet successfully for various node density in the network.

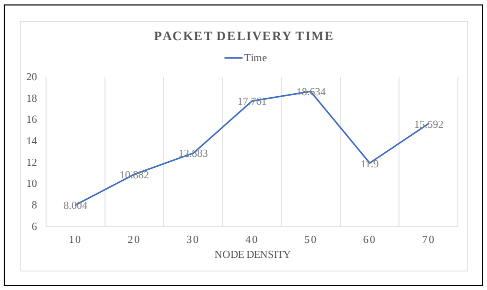


Fig. 4. Packet delivery time vs Node density.

Table 1. Simulation settings

Network Simulator	COOJA under Contiki O.S (2.7)
Simulation time	1 hour
Radio environment	UDGM (Unit Disk Graph Medium)
Area of deployment	600 * 600 m2
Emulated nodes	Sky motes
Transmission Range	40m

4.1 Application domain

As we already mentioned earlier, we have considered the IIoT (Industrial Internet of Things) technology based Additive manufacturing system as our application domain. Now, the primary objective of this project is to secure the transmission between new generation sensors inside of manufactured products that, in particular, will further allow us to analyse the circumstantial data and plan predictive maintenance and process improvement strategies.

In order to achieve this goal the most important requirement is a reliable, robust and secure communication system. Considering the scalability of both the network and generated data, inconsistent and vulnerable nature of sensors and the distributed and heterogeneous network structure, inspired us to think of RPL as the most appropriate solution.

The proposed extension of RPL has several advantages over the existing RPL, such as,

- Every node in the network stores the neighborhood information, which confirms with the distributed nature of the network. This also helps to remove the bottleneck near the root node of every DODAG, especially with a scalable network.
- The degree of routes are dynamically chosen during runtime, ensuring the reliability of the system. Every node selects a set of candidate nodes for forward transmission in such a way, that there always a path towards destination.
- However, if a node selects multiple nodes for next hop transmission and those nodes further repeat the same process, then the control message overhead of the network will increase cumulatively with a chance of duplicate path problem. Hence, our proposed method uses opportunistic route selection paradigm, where the set of candidate nodes will select only one forwarding node, based on the objective functions of the network.
- Since our solution does not involve the root node or common ancestral parent in P2P communication, the time required to transmit a data packet does not depend on the position of the node in the hierarchical structure.

Hence, we can claim that our proposed solution will perform better in this application scenario.

5 Conclusions

The proposed extension over RPL in this paper is responsible for handling the existing issues, such as scalability and mobility in RPL. The proposed routing algorithm is used to identify neighbours of a node that guarantees a reliable transmission towards the destination. The degree of routes is depended directly on the reliability of the process. Thus, the algorithm makes sure that there exist at least one path towards the destination by hopping through nodes: each one finds the best-candidate among its strict neighborhood by guessing the preferred direction. We have used Contiki-Cooja to verify our proposed work. The results clearly show that our method performs better than RPL. The algorithm is parametric over the increment and the width of the rotating angle seeking for such candidate: in our tests we used 10°increments within a 180°maximum span. Future work may focus on making the probability constraint in the condition of the major `if` statement aware of number and density of neighbors.

Acknowledgments. Work partially supported by the project “ADditive Manufacturing & Industry 4.0 as innovation Driver (ADMIN 4D)”, for providing the support required for carrying out the research work.

References

1. T. Winter et. al., ”RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks”, Internet Engineering Task Force (IETF), 6550 Category,ISSN: 2070-1721, March 2012.
2. A. Brandt, J. Buron, G. Porcu, Home Automation Routing Requirements in Low-Power and Lossy Networks, IETF RFC 5826.
3. K. Pister, P. Thubert, S. Dwars, T. Phinney, Industrial Routing Require- ments in Low-Power and Lossy Networks, IETF RFC 5673.
4. M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing Requirements for Urban Low-Power and Lossy Networks, IETF RFC 5548.
5. J. Martocci, P. D. Mil, N. Riou, W. Vermeylen, Building Automation Routing Requirements in Low-Power and Lossy Networks, IETF RFC 5867.
6. M. Chakraborty, N. Deb, N. Chaki., ”POMSec: Pseudo-Opportunistic, Multipath Secured Routing Protocol for Communications in Smart Grid”, 16th International Conference on Computer Information Systems and Industrial Management (CISIM), Lecture Notes in Computer Science, vol 10244. Springer, 2017, Bialystok, Poland, pp. 264-276.
7. M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, ”Security of additive manufacturing: Attack taxonomy and survey”, Additive Manufacturing (21), 431-457, 2018.
8. M. Chakraborty, N. Chaki, A. Cortesi., “A New Intrusion Prevention System for Protecting Smart Grid from ICMPv6 Vulnerabilities”, IEEE Xplore Digital Library proceedings of the 3rd International Workshop on Smart Energy Networks & Multi-Agent Systems (SEN-MAS’14), The Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland, 2014.

9. M. Chakraborty., “Advanced Monitoring based Intrusion Detection System for Distributed and Intelligent Energy Theft: DIET Attack Advanced Metering Infrastructure”, Transactions on Computational Science XXXI. Lecture Notes in Computer Science, vol 10730. Springer, Berlin, Heidelberg, pp. 77-97, 2018, ISSN: 1866-4733, DOI: https://doi.org/10.1007/978-3-662-56499-8_5.
10. Sukho Oh, DongYeop Hwang, Kangseok Kim and Ki-Hyung Kim, ”A hybrid mode to enhance the downward route performance in routing protocol for low power and lossy networks.”, International Journal of Distributed Sensor Networks, 2018, Vol. 14(4), DOI: 10.1177/1550147718772533.
11. W. Brown., ”Distributed Network Architecture: Scalability and load balancing in a secured environment”, White Paper by Tyco Security Products, 2018.
12. W. Gan, Z. Shi, C. Zhang et al., ”MERPL: a more memory efficient storing mode in RP”,19th IEEE international conference on networks (ICON), Singapore, 11-13 December 2013, pp.1-5. New York: IEEE.
13. Contiki, ”Contiki: The Open Source Operating System for the Internet of Things,” 2017. [Online]. Available: http://www.contiki-os.org/.
14. VMware, ”VMware Virtualization for Desktop Server, Application, Public Hybrid Clouds,” 2018. [Online]. Available: http://www.vmware.com/uk.
15. C. Kiraly, T. Istomin, O. Iova et al., ”D-RPL: overcoming memory limitations in RPL point-to-multipoint routing”, IEEE 40th conference on local computer networks (LCN), Clearwater Beach, FL, 26-29 October 2015, pp.157-160. New York: IEEE.
16. Emmanuel Baccelli and Matthias Philipp - The P2P-RPL Routing Protocol for IPv6 Sensor Networks: Tested Experiments - in Software, Telecommunications and Computer Networks.September 2011.