



## Design and Analysis of Low Power and Lossy Network Routing System for Internet of Things Network

**Muhamad Asvial<sup>1\*</sup>**

**Agatha Cracias<sup>1</sup>**

**Muh. Asnoer Laagu<sup>1</sup>**

**Ajib Setyo Arifin<sup>1</sup>**

<sup>1</sup>*Electrical Engineering Department, Universitas Indonesia, Depok – West Java, Indonesia*

\* Corresponding author's Email: [asvial@eng.ui.ac.id](mailto:asvial@eng.ui.ac.id)

---

**Abstract:** In this paper, we create a network for IoT applications that able to reduce power to a minimum. The system uses a combination of objective function zero and the modified E-Trickle (ME-Trickle) algorithm to reduce power usage. The design and simulation are conducted with the Contiki OS, Cooja simulator. The system performance is evaluated based on the parameters of power consumption, packet delivery ratio, latency, and convergence time on the network. The proposed system in the simulation result can reduce the convergence time by 15%, power consumption by 8%, and latency by 9%. This system also continues to provide efficient performance with an average PDR of 93.8%. The simulation concluded that MRHOF provides better performance in network quality, whereas OF0 has a faster network convergence and consumes less power than MRHOF. Therefore, the use of ME-Trickle and OF0 will be appropriate for network scenarios with limited resources.

**Keywords:** IoT, Routing protocol, Objective function zero, ME-Trickle, Power consumption.

---

### 1. Introduction

The Internet-of-Things (IoT) is defined as a “dynamic global network infrastructure with self-configuring capabilities” based on operable standards and protocol of communications. The physical and virtual “things” have virtual identities, physical attributes and personalities, use intelligent interfaces, and are integrated into the information network [1]. The IoT scenario can cover a variety of different applications, such as sensor in urban areas, industrial observations, and home automation [2].

To achieve its goals, the IoT combines several different technologies. It is based on low-power wireless personal area networks, where embedded devices focus on power-saving technologies [3, 4]. To build devices that are energy efficient is to focus on power consumption in the system. Low-power and lossy network (LLN) contains devices with a communication infrastructure that aims to monitor the state of an environment in different locations. However, the weakness of the LLN system is its limited resources because it is small, low of power

and cost. Since almost all sensor use batteries, its means they cannot be recharged, they must work efficiently in terms of computation and communication so that power cannot be wasted [5]. In LLN networks, the system with the highest power consumption is in the transceiver because the nodes in the LLN do not only forward packets to other nodes on the network [6]. Therefore, routing is the biggest concern in wasting resources on this system. The LLN itself has alternative routes from one destination to another, but different quality links. Therefore, a better path selection is required to prevent the problem in routing [6].

The Internet Engineering Task Force has established a routing protocol for the low-power network, called routing protocol for low power and lossy network (RPL), an IPv6 routing protocol for the LLN. RPL is a proactive routing protocol that periodically sends a control message to maintain the network, regardless of data transmission incidents [7]. This protocol uses a destination-oriented directed acyclic graph (DODAG) to create a routing protocol that uses objective function (OF) as a mechanism for choosing the path to the node (root)

that collects all data to the wireless sensor network (WSN) [8]. RPL commonly uses two objective functions: 1) the minimum rank with hysteresis objective function (MRHOF) using the minimum expected transmission count (ETX) metric on the path to the route and the 2) objective function zero (OF0) based on the minimum number of jumps (hops) in the route [9]. Generally, the objective function often used is the MRHOF; however, this does not mean that this OF is the best because it depends on the objectives needed in each scenario.

The previous paper made a comparison of the uses of an objective function in RPL, namely, MRHOF, which is a standard objective function with OF0 [10]. In that study, OF0 obtained a better performance because it produced a lower power consumption than did MRHOF. Ghaleb proposed an algorithm on trickle parameters, called the enhanced trickle (E-trickle) algorithm [11]. E-trickle aims to overcome the short listen problem in the trickle algorithm. The simulation results of this algorithm presented a 43% reduced convergence time and still provided efficiency in power consumption, scalability, and reliability. Meanwhile, another paper conducted a performance comparison of the four variant types of the trickle algorithm, namely, the standard trickle, opt-trickle, E-trickle, and a proposed algorithm, called ME-Trickle [12]. Overall, the best performance was shown by the ME-Trickle in a system with a small density.

The WSN has limited resources; hence, it requires the best possible energy use. The converging time is defined as how long a group of a router can reach the state of convergence. The low convergence time means that it will also reduce power consumption. The more IoT is developed, the more telecommunications devices will be used. No matter how many devices are used, the IoT still has to provide more efficient performance. LLN has several application areas that require both low data rates and high throughput data transfers. It is very important to evaluate the performance of an application. The design of this protocol is influenced by many factors such as energy consumption, network topology, network size, fault tolerance, and latency. Determining RPL standards based on these factors is very important to provide specific performance in the area and cases used. One aspect is in terms of energy efficiency because using large amounts of telecommunication devices will increase the power usage on the network [13]. For that, the network created must be able to reduce power to a minimum.

This article provides a survey of RPL OF enhancement strategies by classifying a couple of

enhancement strategies into strategies based on basic routing metric combination, namely routing metrics specified in RFC [6551] and based on novel mac-oriented metrics and identified the nature of each approach used metrics and combination technique employed [32]. It concludes that OF design is one crucial object that affects routing performances and efficiency, a further study for the ability of these enhancements OF is needed to support network scale and throughput rise.

This article discussed the performance of MRHOF by analysed different ETX hysteresis values [33]. Proposed an optimal value of ETX for the considered network to provide efficient energy and simultaneous reliability for IoT applications. However, using MRHOF conducted the network struggles to obtain all reliable links for different cases in LLN. This paper suggested future work which is to design an OF using different metrics addressing multiple issues.

The present study mainly aims to build a routing system on IoT that focuses on the use of low-power (LLN), that is, it must focus on the convergence time and power consumption. The present study is different from the previous ones [10, 11]. In this system design, the combination of objective function zero and the ME-Trickle algorithm will be used and called RPL OF0-ME. Both modifications largely influence the power consumption of RPL system. The tested network herein contains not only one DODAG system but three DODAG systems at once, because we focus herein on bigger and more complex network system scenarios to see its effect on the network performance [14]. This RPL system can later be used in home automation applications with a sufficiently enough WSN device capacity, such that the power efficiency remains good, although more and more devices are being used [15, 16].

The performance of the design of RPL protocol system will be analysed based on the parameters of convergence time, power consumption, latency, and packet delivery ratio. The analysis of the convergence time and power consumption parameters is conducted to observe the system performance on the power consumption, whereas analysis of the latency and packet delivery ratio (PDR) parameters is conducted to observe the effect of the merging system for power loss on the quality of the network based on other parameters.

The rest of this paper is organized as follows. In section 2 is given an overview of the routing protocol for LLN, section 3 discusses the proposed Design of RPL using OF0 and ME-Trickle, section 4 concludes with system simulation, section 5

contains analysed results and a discussion of performances in each scenario and finally, a conclusion is given in section 6.

## 2. Routing protocol for LLN

There are many types of routing algorithms for IoT [17, 18]. RPL is an IPv6 routing protocol for the LLN that uses a DODAG [31]. This protocol is a distance-vector protocol; hence, it is suitable for LLN. By contrast, the linked state protocol is not suitable for LLN because it requires a significant amount of memory (link-state, link state database (LSDB)). Path selection is very important for RPL. Furthermore, RPL differs from traditional routing network protocols because it uses more factors while computing the best paths (e.g., routing metrics, OF, and routing constraints). A DODAG uses the OF as a mechanism for choosing the path to the node (root) that collects all data to the WSN [10]. OF helps determine if the node is near or far from the root.

Each node on RPL network has a parent option that acts like a gateway for that node. If the node does not have a record on that node, then the node is sent to the chosen parent until it reaches its destination or to the general parent, where it is passed down the tree until it reaches its destination. Nodes on RPL network have their routes in each node. Collecting the route first is not recommended because of some problems in the LLN, including node switching and more losses in the radio and LLN generally do not have a defined topology; thus, RPL must find a link before forming a topology [19]. RPL forms a tree-like topology as shown in Fig. 1 called directed acyclic graph (DAG), with the root at the top and the leaves at the corners. Different from the actual tree topology, RPL provides a backup link, which is required in the LLN. RPL uses the 'up' and 'down' terminology depending on the subject of

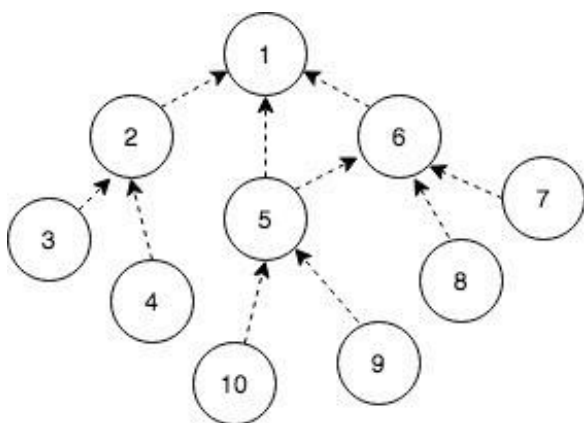


Figure. 1 RPL network topology

traffic. 'Up' refers to the path from the leaf to the root, whereas 'down' is the path from the root to the leaf. The topology information is stored and maintained in the DODAG. The root, which is also called the LLN border router, can be connected to non-LLN networks, such as private networks [20]. In this case, the DODAG is called ground.

The following RPL control messages form the DODAG:

- DODAG information object (DIO): specific nodes in the DODAG can send these messages multicast to inform other nodes, such as when the node is grounded or not, storing or non-storing. DIO messages are used by RPL to form, maintain, and find the DODAG. The DIO contains information about configuring the DODAG to help nodes join the DODAG and select parents.
- DODAG information solicitation (DIS): when no announcement is heard, then a node wants to join the DODAG, and it will send a message if a DODAG is formed. DIS is also used by the node to request DIO messages to the surrounding node.
- DODAG destination advertisement object (DAO): DAO messages are used to propagate the prefix node to the predecessor node to support downward traffic. DAO messages can only be sent when a DODAG is formed.

RPL places a DODAG with parent information. The DODAG uses DIO and DIS to carry the information it owns. The DODAG formation can be arranged in the following regulations [21]:

- path metric,
- OF,
- node policies,
- rules for loop avoidance based on DODAG ranks.

In building a DODAG, the root periodically sends a DIO to the adjacent nodes (neighbours) containing several parameters, such as rank, metric, routing cost, and DODAGID. The neighbouring node determines to join the DODAG based on the OF. The DODAG building process continues when the nodes join the DODAG. After the process is complete, each node has a routing table for each parent doing a hop taking the path to the root node. The node may be sending a DIS, which is used to request information from neighbours while waiting for a DIO message. The process of DODAG forming is shown in Fig. 2. The sink shows the value (Rank= A), which is the value of a node that acts as the root. At the first hop, the rank value will increase to  $(k+1)$  which  $k$  is the redundancy value. Likewise, in the second, third, and so on, the value of  $k$  will be added to the number of hops. Therefore,

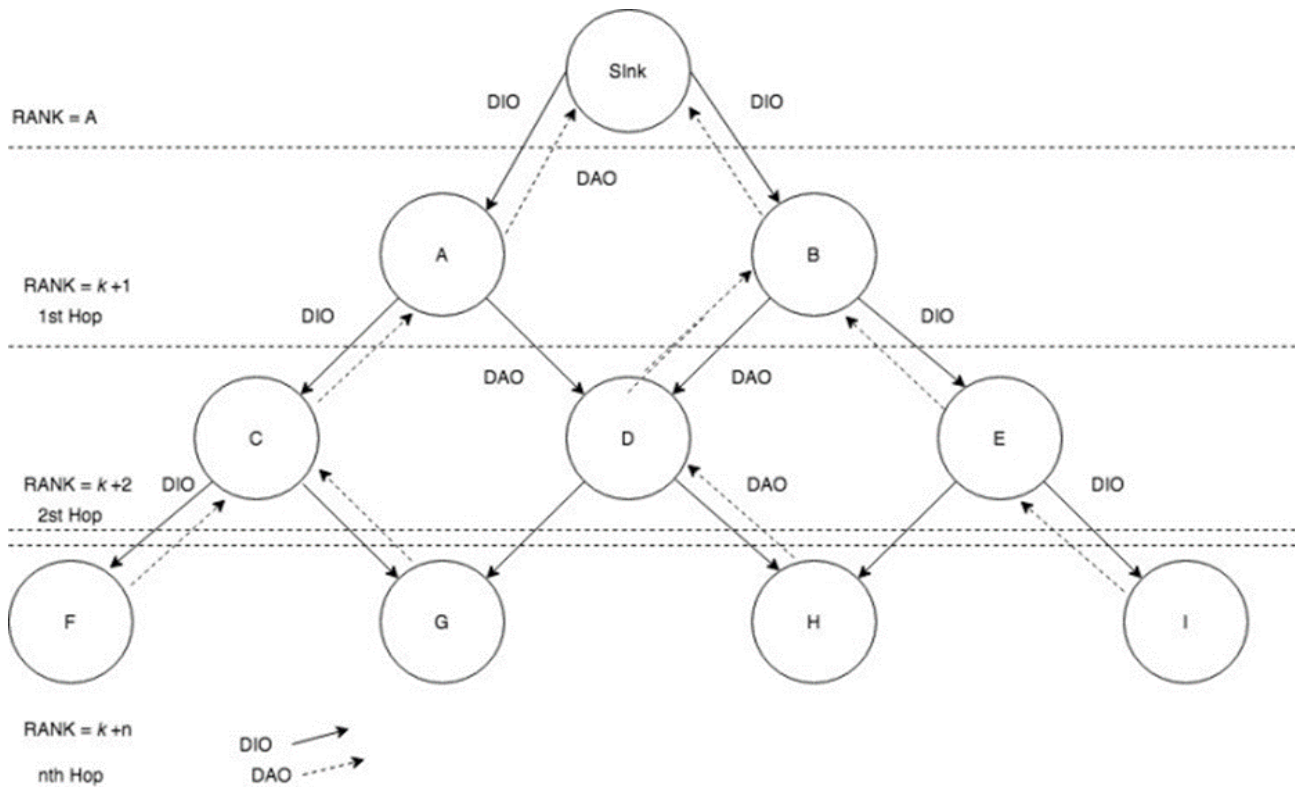


Figure. 2 DODAG forming process

when the hop is farther, the value on the node will be bigger, which means that the rank will be lower. The DAO transmits data based on two modes of operation, namely, storing and non-storing [22]. The DODAG uses a DODAGSequenceNumber to show the latest information [4].

### 2.1 RPL OF

The OF is used to define one or more metrics to help the node translate these metrics to rank. Rank computation is met using the OF depending on the routing metrics as references, such as link quality and delay, and to show the distance between the nodes on the network and the DODAG root. The types of OF usually used for the DODAG are as follows:

- MRHOF uses the minimum ETX (Expected Transmission Count) metric on the path to the route. ETX is the number of times a node can transmit and retransmit packets for successful delivery. The main goal of ETX is to find the route with the highest probability of packet delivery as seen in Eq. (1).

$$ETX = \frac{1}{d_f \times d_r} \tag{1}$$

For example, the measured value of ETX on the link from node A to B. Where  $d_f$  is the forward

delivery rate which is the measured probability value that the acknowledgment is successfully received by node A and  $d_r$  is the reverse delivery rate of the link which is the measured probability value that a packet is successfully received by node B [24].

- OF0 based on the minimum number of hops to the route Eq. (2):

$$R_N = R_p + rank\_increase$$

$$rank\_increase = (R_f \times S_p + S_r) \times MinHopRankIncrease \tag{2}$$

The Rank of a node  $R_N$  is calculated by adding a strictly positive scalar value ( $rank\_increase$ ) to the rank of selected preferred parent  $R_p$ . Rank is considered a fixed number, the position of the radix point between the integer part and the fractional part is determined by  $HopRankIncrease$  shown in [24], Eq. (2).  $MinHopRankIncrease$  is the minimum rank increment between a node and one of its DODAG parents. A DODAG root provides  $MinHopRankIncrease$  that creates a standard between the possible hop cost and the maximum number of hops the network can support. For example, the very large  $MinHopRankIncrease$  allows proper

characterization of a specific hop effect on Rank but cannot support multiple hops [19].

The *rank\_increase* variable is represented in units expressed by the *MinHopRankIncrease* variable, which default to a fixed constant. *step\_of\_rank*  $S_p$  is represents a value related to the parent link metric and properties such as the hop count.  $S_p$  is intermediate computing based on link properties with a certain neighbour, calculated for the link, is multiplied by *rank\_factor*  $R_f$  and then the possibility of being stretched by *stretch\_of\_rank*  $S_r$ .  $S_r$  is a maximum augmentation to *step\_of\_rank* of preferred parent to be able to make viable additional successor selection. *step\_of\_rank* is not stretched if nothing is configured.  $R_f$  is a configurable factor used to multiply the effect of the link properties on the *rank\_increase* calculation. *rank\_factor* of 1 will be used if there is no configured device. The result of *rank\_increase* is added to the Rank of the chosen or preferred parent  $R_p$  to get the rank on the node  $R_N$  [26].

## 2.2 Trickle timer algorithm

One way that an LLN saves energy is by using trickle timers that control the DIO messages to be sent periodically [27]. The duration will double when a message is sent. The trickle algorithm is used to limit the number of control packets sent. When an RPL network is run, DIO messages are transmitted at a value equivalent to  $I_{min}$  and will double every message transmitted until it reaches  $I_{max}$  [4]. The parameters in the trickle algorithm that can be considered by RPL are as follows:

- $I_{min}$ : The minimum value on a trickle timer is called the  $I_{min}$ . This parameter shows the minimum time between two DIOs. The  $I_{min}$  value is determined by minimum interval DIO (*RPL\_DIO\_INTERVAL\_MIN*), and computed as follows Eq. (3):

$$I_{min} = 2^{(RPL\_DIO\_INTERVAL\_MIN)} \quad (3)$$

- $I_{max}$ : The maximum value of trickle timers is used for the number of times the  $I_{min}$  value can be doubled.  $I_{max}$  value is determined from DIO doubling interval (*RPL\_DIO\_INTERVAL\_DOUBLING*) as follows Eq. (4):

$$I_{max} = I_{min} \times 2^{(RPL\_DIO\_INTERVAL\_DOUBLING)} \quad (4)$$

- *Redundancy constant, k*: A number greater than 0 used to suppress DIO transmissions.

Apart from this parameter, there are other variables that also used to control the Trickle timer:

- $I$ : The current interval size.
- $t$ : Random time variable within the current interval.
- $c$ : A counter.

The following six steps comprise the operation of a standard trickle algorithm [11]:

- 1) Trickle starts the first interval by setting the value of  $I$  from the range of values of  $I_{min}$  and  $I_{max}$ .
- 2) When the interval starts, the trickle resets counter  $c$  to 0 and randomly assigns the value in the interval to variable  $t$  from the range of values of  $I/2, I$ . This helps reduce the effects of the listen-only period.
- 3) Trickle raises the counter value to 1 after receiving a “consistent” message.
- 4) The  $t$  value is chosen randomly; hence, trickle presses the scheduled message if the counter is greater or equal to the redundancy constant  $k$ . If it is not bigger or the same, then the message will be transmitted.
- 5) The trickle will double the interval size when the interval  $I$  run out. When the size of the new interval exceeds the  $I_{max}$ , the trickle will set the size of the interval  $I$  to  $I_{min}$  and run the new interval as in Step 2.
- 6) When an “inconsistent” message is detected and has not been set to  $I_{min}$  the trickle will set the value  $I$  to  $I_{min}$  and then start a new interval like that in Step 2.

Consistent message means that there is no change in the received message but “consistent” and “inconsistent” message can also be defined depend on how a protocol itself uses trickle [4].

## 3. RPL using OF0 and ME-trickle

### 3.1 Objective function zero

Based on power consumption, OF0 has better performance because parent selection is based on the number of jumps [10]. However, MRHOF is an OF standard usually generally used in RPL systems. Therefore, in this simulation, an RPL network is designed with OF0. In OF0, each node will calculate the rank based on the value of the jump to the root node. The smaller the number of jumps, the higher the priority node link to be chosen by OF0. The rank value on the child node will always be higher than the parent node because it is getting further away from the root. The change in the use of MRHOF to

OF0 is conducted by using the open-source code `mrhof.c` that contains a parent selection algorithm based on the ETX and the `of0.c` source code containing a parent selection algorithm based on the number of hops modified following research needs.

### 3.2 ME-Trickle

One of the problems with trickle timers is ‘short listen’. Short listen is a problem that occurs mainly because of the asynchronous trickle intervals at each adjacent node. This problem has a very drastic impact on the trickle suppression mechanism, especially on trickle scalability [12]. Short listen occurs in RPL because not all nodes can synchronise, especially because some message nodes do not reach all nodes on the network. Therefore, the node that always multicast first will use too much power, whereas the problem with the LLN has limited resources. Trickle introduces the listen-only period, a period when the trickle does not transmit in the first half of the interval  $l$  [29]. However, this period causes an increase in delay in spreading the transmission intended to resolve the detected inconsistencies, which will negatively affect the time it takes for the network to form, especially when choosing a large  $I_{min}$  value [27]. The longer convergent time will affect the power consumption used, whereas the problem with the LLN has limited resources. Therefore, this change aims to eliminate the listen-only period.

The trickle algorithm has four modification stages. First, change the random variable range  $[I/2, I]$  to  $[0, I]$  to eliminate the listen-only period [10]. Second, rather than resetting the value of  $c \rightarrow 0$  ( $c$  is a counter variable) at each new interval, resetting  $c \rightarrow 0$  will only be placed at the first interval of  $I_{min}$  and during a random timing to eliminate the cumulative impact of the listen-only period. Setting  $c \rightarrow 0$  during the random time selection will produce unequal intervals between the nodes (i.e., the length of the interval in this case). Therefore, a node with a smaller interval length will have a greater chance of transmission.

To overcome this problem, [13] came up with a mechanism to buffer the constant redundancy value  $k$  to make each node have the same opportunity in transmission called *new k*. The solution is to introduce a variable, called  $I_{nz}$ , which is the interval of time difference between the two transmission times. The  $k$  value is reset with the following equation Eq. (5):

$$new\ k = (k \times (2 \times I_{nz} - I))/I \quad (5)$$

From this new  $k$  value, we can see that the greater the  $I_{nz}$ , the greater the  $k$  value. Therefore, if a node has two widely separated transmissions, the  $k$  value will increase such that the redundancy counter  $c$  will be smaller than, and the node will likely be transmitted. After a re-examination, all nodes will be able to resolve consistently between two intervals. Thus, it will likely be more efficient if all nodes directly become maximum intervals when starting a new interval compared with when multiplying these intervals several times when no inconsistencies arise [29]. If at the new interval, the value is changed directly to  $I_{max}$ , and RPL packets sent can be fewer, allowing a reduction in the power consumption. The pseudocode of the trickle algorithm, which has been modified and called ME-Trickle is presented in Fig. 3 [13].

### 4. Simulation

Performance analysis for RPL was conducted on a Cooja simulator. Five algorithms are used in this system: `udp-sink.c`, `udp-send.c`, `rpl-config.h`, `of0.c`,

---

#### Algorithm 1: ME-Trickle

---

```

I. initialization:
   i ← Imin
   c ← 0
II. StartNewInterval:
   i ← Imax
   if Imax ≤ I then
     i ← Imax
   end if
   t ← random(0, I)
III. ReceivedConsistentTransmission:
   c ← c + 1
IV. ReceivedInconsistentTransmission:
   i ← Imin
   c ← 0
V. RandomTimerExpires:
   if inz ≤ i then
     kn ← (k × (2 × inz - i))/I
   else
     kn ← k
   end if
   if c < kn then
     TransmitScheduledDIO
   else
     SuppressScheduledDIO
   end if
   c ← 0

```

---

Figure. 3 ME-Trickle pseudocode

and `mrhof.c` distributed as open-source code modified in accordance with the purpose of the present study. Cooja has a collect view graphical user interface (GUI); thus, the necessary parameters can be indicated. The parameters required for this simulation are shown in Table 1.

#### 4.1 Network setup

The present study simulates a full network system with three sink nodes and a border router that works as a root to form a DODAG system, such that three DODAG systems are formed with each DODAG having 30 sender nodes.

#### 4.2 Metric evaluation

The collect view GUI has a power consumption parameter. Power consumption is the power used by each node and measured in the network during a simulation. It consists of four types of power measurements: low-power mode (LPM), CPU power, radio listen power, and radio transmit power. The power consumption was measured herein by the accumulation of these four types of power. The GUI also showed the number of packets that arrived and the number of packets lost. These data were used to reduce the packet delivery ratio. The radio log on the Cooja simulation contained the altitudes of each node during the simulation based on time. This log also contained information about each message sent by the system such as the type of message, the origin and destination of the message being sent at a certain time.

Table 1. Simulation parameters

Parameter	Value
Radio model	UDGM
Hardware platform	Skymote
Transmission/ interference range	50m/ 55m
Transmission ratio Tx(%)	100
Reception ratio Rx(%)	60 - 100
Node placement	Rand-om
Objective function	OF0
Total sink node	3
Total sender node	90
Simulation time	8 min/ 480s
Area range	100 x 100
DIO min	12
DIO doubling	8
Send interval	4
Trickle redundancy counter, k	10
RDC channel check rate	16

#### 4.3 Metric performance measurement

Four standard performance metrics were used in the simulation:

- Network Convergence time  
The convergence time in RPL DAG is defined as the amount of time needed by the node that can be reached (i.e., radio in this case) in the network to join the DAG. This convergence is considered as the first convergence time on RPL network with a static code. However, some nodes are mobile and have lossy links; thus, the convergence time value is not absolute on the wireless network in an LLN. The convergence time on RPL network is determined by the predetermined time for the first DIO to be sent from the client node and the last DIO to join the DAG. The convergence time is obtained from the following equation Eq. (6):

$$\text{Convergence Time} = \text{Last DIO joined DAG} - \text{first DIO sent} \quad (6)$$

The variable data for the time when the DIO was first sent until the DODAG was formed were contained in the Cooja radio log containing the node activity record when sending RPL setting messages.

- Power consumption  
The radio on-time percentages on the radio was used for an accurate estimation in terms of power consumption. The radio on time comparison on the radio ware used because the radio dominates the power usage in sensor nodes. The radio percentage is the average of all nodes on the entire network. A power trace system on Contiki was used to measure power consumption. A power trace system measures low-power wireless network power usage levels such that the power consumption from the CPU, packet transmission and listening can be estimated. The radio on the time percentage is calculated based on this computation. The power consumption is computed for transmitting and listening to the radio because they consumed the largest amount of power. Power consumption comprised four types of power measurements: LPM, CPU power, radio listen power, and radio transmit power. The power consumption was measured herein by accumulating these four types of power. The variable data of these four power factors are shown in the Cooja collect view in milliwatts.

- Latency

Latency is defined as the amount of time required by a packet from a node to reach a sink. The network latency can be computed as follows Eq. (7):

$$Total\ Latency = \sum_{k=1}^n (recvTime(k) - SentTime(k)) \quad (7)$$

Where  $n$  is the total number of packages that successfully received,  $recvTime(k)$  is the time when the package was successfully received and  $SentTime(k)$  is the time when the node starts sending the packet. The information on sending and receiving times is shown on the simulator cooja via the radio log that records every node activity. The average latency is taken from the sum of all latencies, with the number of packets successfully received. The average latency is also shown in the radio log Cooja conclusion function. The average latency is calculated as follows Eq. (8):

$$Average\ Latency = \frac{Total\ Latency}{Total\ Paket\ Received} \quad (8)$$

- Packet delivery ratio

PDR is defined as the ratio of the number of packets received in a sink to the number of packets sent to a sink. The PDR taken is the average of packets successfully received in the sink from all nodes. The PDR average calculation is conducted by comparing the packets sent to all nodes to the sink with the number of packets successfully received in the sink [30]. The PDR variable obtained comes from the collect view of received packet data and lost packet data [eq. (9)]. The total number of packets received by the root of all nodes is summed by the total number of packets lost from all nodes as the total number of packets sent.

$$Average\ PDR = \left( \frac{total\ received\ packet}{total\ packet\ sent} \right) \times 100 \quad (9)$$

#### 4.4 Simulation system and topology

The simulation was conducted in several stages to see the process of improving RPL system performance. Furthermore, the simulation was tested based on three system scenarios: standard RPL system with MRHOF and the original trickle timer algorithm (i.e., RPL MRHOF); RPL system using

OF0 but still on the original trickle timer algorithm (i.e., RPL OF0); and RPL system using OF0 and ME-Trickle, called RPL OF0-ME. Each simulation scenario lasted for 8 min. When the simulation started, the collect view started to collect data, and the radio log recorded each node's activity based on time.

In this system, three nodes act as the root or border router. All these roots are close to each other; hence, they can still be connected. However, different sender nodes DODAG are at great distances, so they are not connected to each other which means that each sender node is only connected to one DODAG system even though there are three DODAG systems in this network. Only three sink nodes were used because in RPL system standard, only one sink node works as the root of each DODAG. Meanwhile, each DODAG system has 30 sender nodes because the number is not too big but not small either for a density node.

The simulations were conducted at different Rx ratios of 60% to 100% for each scenario. The design topology with 10 m range in each grid in Fig. 4 below:

## 5 Result and discussion

### 5.1 Performance analysis based on the convergence time

This section shows the results of the convergence time in the simulation of RPL MRHOF, RPL OF [11], and RPL OF0-ME. The graph of the comparison of convergence time with Rx values on the three networks as shown in Fig. 5 (a):

The graph showed that when the Rx ratio condition was 70% – 100%, no significant difference can be found in the time convergence of the three systems. However, RPL OF0-ME system had a lower convergence. The difference between the three systems was clearly seen when the Rx ratio was 60% because RPL OF0-ME was not affected by the short listen problem and the listen-only period, which is an issue in RPL system standard.

In RPL OF0-ME, the elimination of the listen-only period occurs when the transmission time of the random  $t$  variable can be started without having to wait for the first half of the interval. This allows nodes to compete at wider interval sizes to reduce the possibility of collisions. An elimination occurs because the redundancy counter,  $c$ , is reset only when the random time has been determined, not when each interval starts to decrease the cumulative impact in short listen problems. The random selection of time starting from the beginning of this



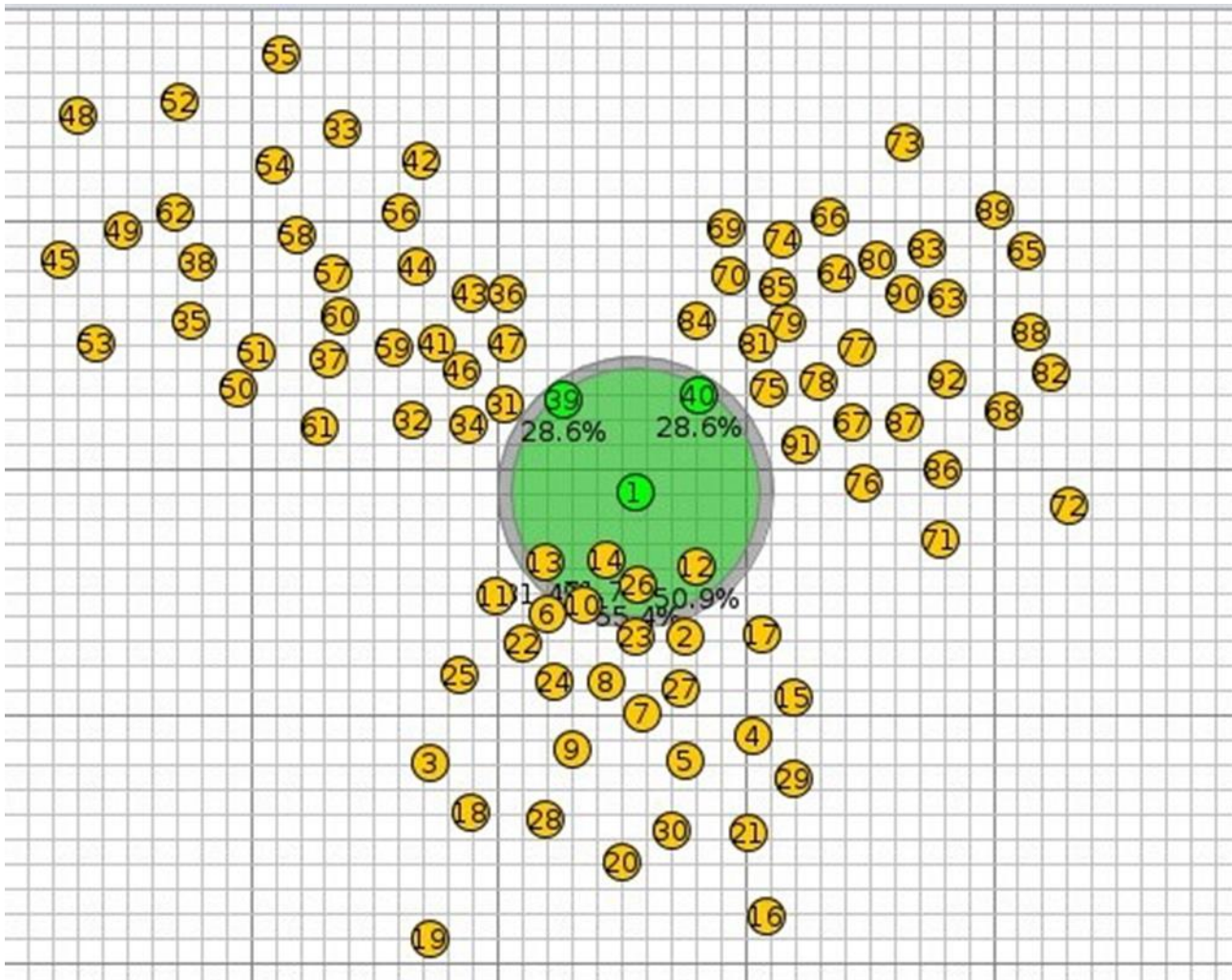


Figure. 4 RPL network topology design

interval causes a decrease in delay, which means the network will form faster than the standard RPL system. The convergence time decreases because the network formation process is faster. In RPL OF0-ME, the resulting performance is quite good because it can reduce the convergence time.

## 5.2 Performance analysis based on power consumption

This section shows the comparison of power consumption in each simulation as shown in Fig. 5 (b). The evaluation of the power consumption performance showed that both the ratios of 100 to 60 from RPL OF0-ME system had a lower power consumption than the other two systems. This can happen because MRHOF-ETX prefers long links with a more stable quality than short links with poor quality, such that many nodes in routing packets are stacked and through longer paths, which requires a higher power consumption. Additionally, the

algorithmic calculation of the ETX metric for MRHOF is more complex than OF0 based on the node rank. Additionally, OF0 is based on minimizing the number of jumps to the root, resulting in reduced data packet retransmission and radio use. Higher power consumption is used on radio listen power in both objective functions because of the number of packets circulating on the network. Therefore, each node consumes more power when radio listen to the channel to send packets because it reduces collisions.

The decrease in power consumption is attributed to the use of the ME-Trickle algorithm. ME-Trickle affects the convergence time because the network formation with it is faster due to the listen period of RPL standard, which requires transmission only after the first half of the interval is eliminated. The length of time of network formation is directly proportional to the power consumption; hence, the lower the convergence time, the lower the power consumption. Using the ME-Trickle algorithm will

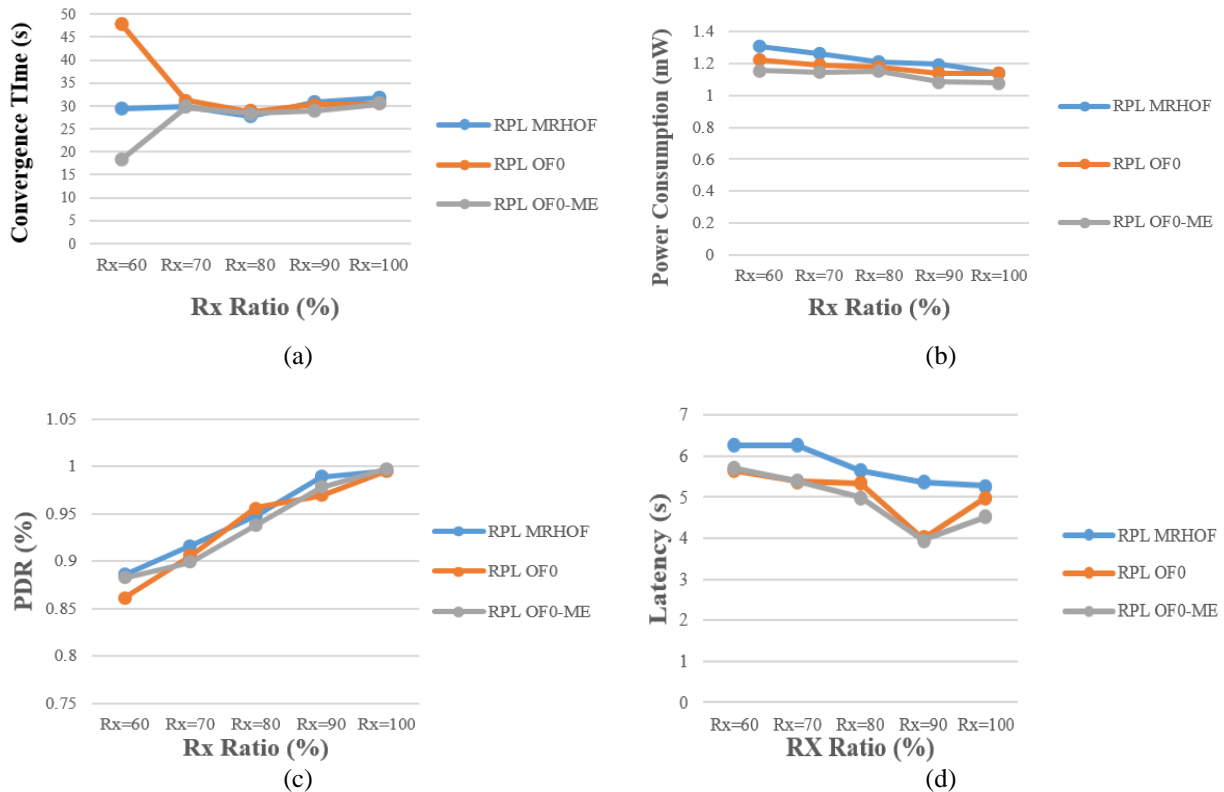


Figure. 5: (a) comparison graph of convergence time simulation results on RPL MRHOF, RPL OF0, and RPL OF0-ME against Rx, (b) comparison graph of power consumption simulation results on RPL MRHOF, RPL OF0, and RPL OF0-ME against Rx ratio, (c) comparison graph of the PDR simulation results on RPL MRHOF, RPL OF0, and RPL OF0-ME against the Rx ratio, and (d) comparison graph of the latency simulation results on RPL MRHOF, RPL OF0, and RPL OF0-ME against the Rx ratio

indirectly reduce power consumption. In the sub-interval procurement process, the standard trickle algorithm assigns a doubling value to two in each case, regardless of the node requirements. No inconsistency is observed when inspecting packages with intervals. This causes excessive power usage because the procurement is done many times. In the ME-Trickle algorithm, the interval is not duplicated for each expired sub-interval; instead, the expired interval is immediately converted to an  $I_{max}$  value. The conversion results in fewer RPL messages being sent or transmitted. This less packet transmission causes the power consumed to decrease. Therefore, in the ME-Trickle algorithm, power consumption can be lower, but the problem of inconsistent messages can still be overcome. Eliminating the listen-only period on RPL system results in the emergence of the short listen problem on RPL such that the node is out of sync and over-transmits. However, this can be overcome by the ME-Trickle algorithm by stretching the k value in accordance with the value of the vulnerable variable two-time intervals. In this way, the node knows the right time for transmission, and each node has the same opportunity of transmission.

### 5.3 Performance analysis based on packet delivery ratio

The PDR is the packet ratio successfully reached. This section will show the simulation results of three systems with different Rx ratio conditions. The average packet sending ratio from the node to the border router of each packet for 8 min of simulation is shown in Fig. 5 (c):

The graph shows that the PDR performance of each system is almost balanced when the Rx ratio is 100%. This is because the Rx ratio of 100% is the ideal value, and it is very likely that there will be no packet loss. The other ratio values do not change significantly. By contrast, the PDR value of RPL MRHOF and RPL OF0-ME have almost the same results. Generally, the effects of the changes in the OF and the trickle algorithm on the PDR are not very significant. In other words, a system with RPL OF-ME still provides the same effectiveness after being modified. Comparisons of the power consumption and the PDR, showing that power consumption is inversely proportional to the PDR performance as shown in Figs. 5 (b) and (c). When

the Rx ratio became smaller, the PDR also became smaller, but the power consumption became higher. This happened because when the PDR was low, the system continued to send control packets until the best route was determined, increasing the energy consumption.

#### 5.4 Performance analysis based on latency

This section shows the simulation results for latency. The differences in the latency of each system are analysed herein. The analysis was taken from the average latency of each packet for the simulation duration.

The latencies of RPL OF and RPL OF0-ME were much lower than that of RPL MRHOF as shown in Fig. 5 (d). The most obvious decrease was seen in the 90% – 100% ratio. As previously explained, OF0 routing is done by choosing the fewest jumps; therefore, compared with MRHOF, the latency will be lower. The route with fewer jumps made the packet sent to arrive faster. In other words, the change in the modification of the routing path selection with OF0 decreases not only the power consumption but also the latency. RPL OF and RPL OF0-ME did not have significant differences, but RPL OF0-ME's performance in terms of latency was slightly lower. This result indicates that the change of the trickle algorithm did not affect the latency but still provided the same performance as the system before it was modified. This change in the algorithm was more focused on changing the network formation process and did not affect how long the packets are sent before and after the network is formed. Furthermore, as previously explained, the modified trickle algorithm shows the best performance when it is at the least node or low density. In this system, each DODAG only has around 30 nodes, which means it is not a large density.

## 6 Conclusion

The present study designed a routing system for IoT applications with the routing protocol for an LLN using the OF zero and ME-Trickle algorithm. The design was done on Contiki OS software. The performance evaluation was analysed through four parameters, namely, power consumption, converging time, PDR, and latency. The simulation was run on a Cooja Simulator, version 2.7. The simulation conducted was used to analyse RPL OF0-ME system, which provides better performance than the OF standard and the previous trickle algorithm. RPL OF0-ME system can reduce the power consumption by up to 8%, the convergence

time by up to 15% and the latency by 9% compared with the existing RPL standard. This modified system also continues to provide efficient performance with an average PDR of 93.8%. The most noticeable increase was observed in the convergence time when the Rx ratio was low, power consumption and latency. The simulation concluded that MRHOF provides better performance in the aspect of network quality, whereas OF0 has a faster network convergence and consumes less power than MRHOF. Therefore, OF0 will be appropriate for network scenarios with limited resources.

#### Conflicts of Interest

The authors declare no conflict of interest This article has not been published and is not under consideration for publication elsewhere. The authors have no conflict of interest in regard to this research or its funding

#### Author Contributions

The first author, Muhamad Asvial is contributed to the conceptualization, methodology, implementation of algorithms, the conduct of experiments, and the formation of the paper. The second author, Agatha Gracias is contributed to writing-original draft preparation, resource data, and software simulation. The third author, Muh. Asnoer Laagu is contributed to analyse data, supervision and checking the paper. The fourth author, Ajib Setyo Arifin contributed in proofreading the paper.

#### Acknowledgments

The authors thank to Universitas Indoensia for supporting funding for this research. The funding is the assignment grant research Universitas Indonesia 2021.

#### References

- [1] I. G. Smith, ed. *The Internet of things 2012: new horizons*, 2012.
- [2] M. Asvial and M. A. Laagu, "New Development of Physarum Routing Algorithm With Adaptive Power Control", *IEEE Access*, Vol. 9, pp. 74868-74878, 2021,
- [3] S. Budiyo, M. Asvial, and D. Gunawan, "Implementation of genetic zone routing protocol (GZRP) in 3G-WiFi offload multi base station", In: *Proc. of 2013 IEEE International Conference of IEEE Region 10*, pp. 1-6, 2013.

- [4] M. Asvial, S. Budiyo, and D. Gunawan, "An intelligent load balancing and offloading in 3G—WiFi offload network using hybrid and distance vector algorithm", In: *Proc. of 2014 IEEE Symposium on Wireless Technology and Applications (ISWTA)*, pp. 36-40, 2014.
- [5] M. A. Laagu and M. Asvial, "The Physarum Algorithm with Adaptive Power Control: A Routing Optimization to Overcome Obstacle Problems in IoT Network", *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 1077, No. 1, p. 012054, 2021.
- [6] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm", *IETF RFC 6206*, 2011.
- [7] L. Rosyidi, M. Asvial, and R. F. Sari, "Efficient router node deployment for ZigBee based Internet of Things network using Physarum optimization algorithm", In: *Proc. of 2017 International Conference on Information and Communication Technology Convergence*, pp. 639-644, 2017.
- [8] M. A. Mahmud, A. Abdelgawad, and K. Yelamarthi, "Improved RPL for IoT Applications", In: *Proc. of 2018 IEEE 61st International Midwest Symposium on Circuits and Systems*, pp. 988-991, 2018.
- [9] A. Dhumane, R. Prasad, and J. Prasad, "Routing issues in internet of things: a survey", In: *Proc. of International Multiconference of Engineers and Computer Scientists*, Vol. 1, pp. 16-18, 2016.
- [10] H. Lamaazi, N. Benamar, M. I. Imaduddin, and A. J. Jara, "Performance assessment of the routing protocol for low power and lossy networks", In: *Proc. of 2015 International Conference on Wireless Networks and Mobile Communications*, pp. 1-8, 2015.
- [11] A. E. Hassani, A. Sahel, and A. Badri, "Impact of RPL objective functions on energy consumption in IPv6 based wireless sensor networks", *Colloque sur les Objets et systèmes Connectés*, 2019.
- [12] B. Ghaleb, A. Al-Dubai, and E. Ekonomou, "E-trickle: Enhanced trickle algorithm for low-power and lossy networks" In: *Proc. of 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1123-1129, 2015.
- [13] D. J. Benson, "A performance study of RPL with trickle algorithm variants", *Worcester Polytechnic Institute*, 2016.
- [14] A. L. Colina, A. Vives, A. Bagula, M. Zennaro, and E. Pietrosemoli, *IoT in 5 days*, 2015.
- [15] M. Asvial, M. F. G. Pandoyo, and A. S. Arifin, "Internet of Things Solution for Motorcycle Riders to Overcome Traffic Jam in Jakarta Using EBkSP", In: *Proc. of 2018 International Conference on Information and Communication Technology Convergence*, pp. 636-638, 2018.
- [16] M. Frey, F. Große, and M. Günes, "Energy-aware ant routing in wireless multi-hop networks", In: *Proc. of 2014 IEEE International Conference on Communications*, pp. 190-196, 2014.
- [17] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", *IEFT. RFC 5826*, 2010.
- [18] Y. Tian and R. Hou, "An improved AOMDV routing protocol for internet of things", In: *Proc. of 2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2010.
- [19] S. M. A. Oteafy, F. M. Al-Turjman, and H. S. Hassanein, "Pruned adaptive routing in the heterogeneous Internet of Things", In: *Proc. of 2012 IEEE Global Communications Conference*, pp. 214-219, 2012.
- [20] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. K. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", *IEFT. RFC 6550*, 2012
- [21] S. Duquennoy, F. Österlind, and A. Dunkels, "Lossy links, low power, high throughput", In: *Proc. of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 12-25, 2011.
- [22] J. P. Vasseur and A. Dunkels, "Interconnecting smart objects with ip: The next internet", *Elsevier Science & Technology*, 2010.
- [23] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of RPL routing protocol", In: *Proc. of 2011 IEEE International Conference on Mechatronics*, pp. 767-772, 2011.
- [24] C. Houaidia, A. Van Den Bossche, H. Idoudi, T. Val, and L. A. Saidane, "Experimental performance analysis of routing metrics in wireless mesh networks", In: *Proc. of 2013 9th International Wireless Communications and*

- Mobile Computing Conference*, pp. 1011-1016, 2013.
- [25] J. P. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks", In *RFC 6551*, pp. 1-30, 2012.
- [26] O. Iova, F. Theoleyre, and T. Noel, "Exploiting multiple parents in RPL to improve both the network lifetime and its stability", In: *Proc. of 2015 IEEE International Conference on Communications*, pp. 610-616, 2015.
- [27] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (RPL)", *IETF. RFC 6552*, 2012.
- [28] B. Djamaa, and M. Richardson, "Optimizing the trickle algorithm", *IEEE Communications Letters*, Vol. 19, No. 5, pp. 819-822, 2015.
- [29] L. Kalyan and K. Kaur, "An Analysis of Performance of Routing Protocol for Low Power and Lossy Network in Sparse Wireless Sensor Network", *International Journal of Advanced Research in Computer Engineering & Technology*, Vol. 4, No. 8, 2015.
- [30] P. Levis, N. Patel, D. Culler, and S. Shenker. "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks", In: *Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation*, Vol. 25, 2004.
- [31] C. Guo, J. Zhou, P. Pawelczak, and R. Hekmat, "Improving packet delivery ratio estimation for indoor ad hoc and wireless sensor networks", In: *Proc. of 2009 6th IEEE Consumer Communications and Networking Conference*, pp. 1-5, 2009.
- [32] I. Kechiche, I. Bousnina, and A. Samet, "An Overview on RPL Objective Function Enhancement Approaches", In: *Proc. of 2018 Seventh International Conference on Communications and Networking*, pp. 1-4, 2018.
- [33] M. A. Mahmud, A. Abdelgawad, and K. Yelamarthi, "Improved RPL for IoT Applications", In: *Proc. of 2018 IEEE 61st International Midwest Symposium on Circuits and Systems*, pp. 988-991, 2018.