

Enhanced mobility aware routing protocol for Low Power and Lossy Networks

Shridhar Sanshi¹  · C. D. Jaidhar¹

Published online: 11 November 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

Abstract Due to the technological advancement in Low Power and Lossy Networks (LLNs), sensor node mobility becomes a basic requirement for many extensive applications. Routing protocol designed for LLNs must ensure real-time data transmission with minimum power consumption. However, the existing mobility support protocols cannot work efficiently in LLNs as they are unable to adapt to the change in the network topology quickly. Therefore, we propose an Enhanced Routing Protocol for LLNs (ERPL), which updates the Preferred Parent (PP) of the Mobile Node (MN) quickly whenever the MN moves away from the already selected PP. Further, a new objective function that takes the mobility of the node into an account while selecting a PP is proposed. Performance of the ERPL has been evaluated with the varying system and traffic parameters under different topologies similar to most of the real-life networks. The simulation results showed that the proposed ERPL reduced the power consumption, packet overhead, latency and increased the packet delivery ratio as compared to other existing works.

Keywords Low Power and Lossy Networks · Distance · Objective function · Preferred Parent · Routing protocol

1 Introduction

Internet of Things (IoT) is one of the rapidly developing technologies, which aims to connect things around the world [1]. Wireless Sensor Network (WSN) is an important paradigm of IoT, which consists of several sensor nodes. These nodes sense the surrounding environment and gather the application-specific information, and then send to the collector node over the Internet for processing purpose. There are many applications of WSN, which are gaining popularity in the real-world, like in military, environment monitoring, medical applications, health care etc. [2–5]. However, sensor nodes of the WSN are characterized by small size, battery powered with low computing capability. Designing a routing protocol for such a network is a challenging task [5].

In order to route packets over the wireless transmission link, Internet Engineering Task Force has created the Routing Over Low power and Lossy Networks (ROLL) working group and later ROLL working group come up with IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [6], which is capable of fulfilling the precise requirements of Low power and Lossy Networks (LLNs). These networks are characterized by their low capacity links, high loss, and low power requirements. WSN is a part of LLNs [7]. RPL is genuinely a simple distance vector routing protocol designed for LLNs, it constructs Directed Acyclic Graph (DAG) rooted towards DAG root using the control messages and Objective Function (OF). Routes constructed using OF are optimized routes to the destination and RPL maintains multiple routes to the destination. The OF uses a routing metric to construct DAG, the default routing metric used in RPL is Expected Transmission Count (ETX) [8], which is the

✉ Shridhar Sanshi
it15f03.sanshi@nitk.edu.in

C. D. Jaidhar
jaidharc@nitk.edu.in

¹ Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Karnataka, India

expected transmissions required by the node to accomplish the task of delivering the packet to its destination.

Initially, RPL is designed with an aim of working for the static topology. However, due to the popularity of LLNs many applications like industrial automation, warehouse, healthcare monitoring etc. require mobility support and most of the applications demand transmission of data in real-time [9]. Hence, RPL protocol should support for mobility and provide guaranteed reliability to transmit data during mobility of the node. RPL protocol suffers due to the movement of the Mobile Nodes (MNs), as their neighboring nodes and the link quality with these nodes constantly change. If the MN is effectively distant from its Preferred Parent (PP) then the route towards the PP becomes inconsistent and continues to exist until route expires. All the packets transmitted to the PP which are out of communication range will be lost and require retransmission of packets to reach the destination successfully.

Most of the literature work is concentrated on varying amount of control information disseminated in the network to accomplish fast identification of the parent node by modifying trickle algorithm [10–13]. However, the methods proposed by previous works take some amount of time to identify whether the MN is in the communication range of the PP or not and takes time to update its new PP. In order to overcome this problem, we propose an Enhanced Routing Protocol for LLNs (ERPL) which is a simple and powerful protocol for the LLNs.

Taking the characteristics of RPL into consideration, the main contributions of this paper are as follows:

- In contrast to previous work, that focused on modifying trickle algorithm, this paper proposes ERPL that updates the PP of the MN quickly whenever MN moves away from the current PP.
- In order to support mobility in RPL, we have designed, implemented, and evaluated an OF that considers multiple routing metrics to select a PP.
- Our experimental results demonstrate that ERPL outperforms as compared to existing works in terms of improved Packet Delivery Ratio (PDR), less power consumption and reduced end-to-end delay.

This paper is structured as follows: in Sect. 2, we present an overview of RPL protocol, then, we discuss motivation and related work on mobility support in RPL in Sect. 3, in Sect. 4, proposed ERPL is presented. Section 5, evaluates the performance of ERPL using the Cooja simulator. In Sect. 6, we discuss the cost (shortage) of the ERPL. Finally, we conclude our proposed work in Sect. 7.

2 Overview of RPL protocol

RPL is a simple, IPv6 distance vector routing protocol designed specifically for LLNs. It organizes the network like a tree structure known as DAG, the tree is rooted towards specific node called as DAG root constructing the Destination Oriented DAG (DODAG). Each of the DODAG is identified by the DODAG Identifier, and are optimized based on the OF in use. In DODAG, a node can associate with more than one node (parent) to establish a route towards the root node. Nodes which have potential to become a parent are selected as candidate parents. The choice of selecting the best parent among candidate parents towards the root node is done based on the OF, which depends on routing metrics or constraints among other candidate parents. In order to select the PP among candidate parent nodes, RPL computes rank based on the OF using control messages. RPL supports four control messages for the construction of DODAG. These control messages are discussed below.

- DODAG Information Object (DIO) message: Initially, DODAG root broadcast the DIO message to start the construction of DODAG, and later the intermediate node broadcast the DIO message to maintain the DODAG. The DIO message consists of all necessary information to compute the rank along the path towards the root node.
- DODAG Information Solicitation (DIS) message: If the node is not part of any DODAG and is not heard any DIO message, then the node broadcast DIS message to receive the DIO message from its neighbors.
- Destination Advertisement Object (DAO) message: The control message used to construct downward route between non-root nodes. Each node sends DAO message to its PP in order to record reverse paths.
- DAO Acknowledgment (DAO-ACK): Message sent to recognize the reception of a DAO message.

DODAG construction begins when the DODAG root broadcast DIO message. Neighbors of the DAG root compute their rank using the information in the DIO message and decide whether to join the DODAG. Then they broadcast their DIO message to their neighbors. This process repeats until DIO messages reach the leaf nodes. For sending a packet to DODAG root, the node selects PP based on the defined OF. A node broadcast DIO message based on the trickle algorithm [14], Initially, trickle timer is set to [Imin], if the node receives consistent information then trickle increases transmission time by 1 up to [Imax] otherwise it resets the timer to [Imin] and starts sending DIO messages more quickly. For more detailed explanation of trickle algorithm refer [14].

3 Motivation and related work

Routing protocol designed for wireless links have to account for changing link qualities in terms of individual links as well as the end-to-end path in dynamically changing environments [15]. Since LLNs consists of wireless links, developing an energy efficient routing protocol is a challenging task. RPL protocol is designed mainly for routing packets over the LLNs for static networks. Adding mobility support to it raises many issues that have to be dealt with.

- Retransmission of packets: Due to a node movement, if a node is effectively distant from its PP then route towards the PP becomes inconsistent and retransmission of packets become evident.
- Control packet overhead: Due to node movement, the trickle timer is reset to increase the frequency of the control messages.
- Inconsistent neighbors: Due to movement of the node, its neighbors position change frequently. If the neighbor's list is not updated in time then the list becomes inconsistent.

In the literature, many research works have attempted to support mobility in RPL. However, solutions are not optimal. As a conclusion, mobility support for the RPL is important and is highly essential. We discuss research work related to mobility support in RPL in rest of the Section.

In [10], the authors have proposed a mechanism for WSN in which the router node learns about the MN from DIO message. While selecting a PP, higher priority is given to the router node than the MNs. The trickle timer is modified dynamically according to the node behavior i.e., if the node changes PP frequently then the DIS interval reduces further. However, the framework is less responsive, whenever the MN changes location suddenly. Therefore, does not prevent from packet loss.

In [11], the authors have proposed reverse trickle algorithm to support the movement of the node seamlessly in the network. The MN advertises its status to the parent node by setting a flag bit in the DAO message. The reverse trickle algorithm is based on the assumption that, initially the MN likely to remain connected to the parent node for a long time. Then after spending more time, more it is likely to move outside the coverage of the parent node. However, the assumption may not be true all the time.

In [12], the authors have modified trickle timer to handle changes in the network topologies. The main idea is to adjust the DIO message interval based on the distance and velocity of the node. To compute the distance, the node sends position information through control messages, and distance is computed using Euclidean distance formula. Simulation results showed that proposed technique

increased the PDR. However, the protocol requires more control messages whenever the MN moves out of the PP communication range. Therefore, it suffers from delay to update the PP. Similarly, in [13], the authors have proposed a modified energy efficient RPL where velocity and angle between the PP and the MN are calculated based on the Doppler frequency. The time interval to select the new parent node is dynamically set based on the distance and speed. However, the protocol experiences some delay to update PP whenever the MN moves out of the PP communication range.

In [16], an additional layer called mobility support layer (MoMoRo) is introduced between the link layer and routing layer to increase the packet reception ratio. If the node is not reachable, then the MoMoRo makes an attempt to reach the node or finds a new path to deliver the remaining packets. MoMoRo uses fuzzy estimators to select the communication links which are good and usable. Since additional layer introduced and frequent collection of neighbor information radically increases the packet overhead.

In [17], authors have analyzed the behavior of RPL under mobility in vehicular ad-hoc networks and proposed an improved RPL. The main contribution includes immediate fire of ETX when a new neighbor node is discovered without scheduling ETX probe and broadcast DIO message immediately whenever PP changes. They also proposed a mechanism for loop detection and avoidance by not allowing the node to join a parent node having rank less than itself. However, the modified protocol is intended for vehicular ad-hoc networks, where the node changes the location very rapidly and also assumes that all the nodes in the network are equal.

In [18], Mobility-Triggered RPL (MTRPL) is proposed. It is a cross-layer protocol lying within link layer and network layer. MTRPL provides better access to transmission resources for the mobile devices by taking an advantage of X-Machiavel preamble sampling medium access control layer protocol [19]. In X-Machiavel, MNs are allowed to steal medium from the ongoing transmission to send their own packet first.

In [20], the authors have introduced Kalman positioning based RPL to provide robust and reliable routing. It selects link between anchor and MN based on the end-to-end ETX values. The end-to-end estimates are obtained using Kalman filter to reduce the impact of channel dynamics. However, the mechanism incurs a lot of processing overhead for the MN.

In [21], the authors have proposed an extension to the standard RPL which uses corona mechanism to support the mobility (CO-RPL). The basic idea is to create coronas around DAG root, which allows localization of the RPL routers in motion. The node selects PP based on the corona ID and rank of the parent node to prevent loop in the network. However, even though the protocol increases

PDR, it consumes more power and increases control packet overhead in the network.

In [22], the authors have proposed a framework for handoff process along with a SmartHop algorithm to support mobility for lossy networks. The smarthop algorithm works in data transmission phase and discovery phase. In data transmission phase, serving access point sends a reply with the average Received Signal Strength Indicator (RSSI) after receiving ‘ n ’ number of packets within a window from the MN. The MN monitors the link quality from the received message sent from the access point. MN selects its PP based on the RSSI and parent node that does not lie in the same sub-DAG. The simulation results showed better PDR. However, it suffers from delay in updating PP.

Recently, in [23], the authors have proposed a cluster-based routing (SMART) scheme for the cognitive radio network. The SMART technique uses reinforcement learning, an artificial intelligence approach. The cluster based approach enhances network stability and improves the network performance by selecting more stable routes and reduce the frequency of route discovery.

4 Enhanced RPL for mobile network

ERPL is proposed to provide mobility support to the standard RPL, which includes the following modifications:

- MNs are not allowed to transmit DIO messages, hence, they are not allowed to become PP, which reduces some amount of retransmissions and control packet overhead.
- OF of the Fixed Router Node (FRN) uses multiple routing metrics i.e. ETX, Residual Energy (RE) and Distance (Dist) which ensures that the link is of high quality, high availability and consumes less power.
- OF of the MN uses multiple routing metric i.e. ETX, Dist, and Distance Variation (Dist_Var). Dist is given more priority over other two metrics which ensures that the MN selects the nearest PP in the direction of the movement.
- Frequently updating neighbors to avoid using the inconsistent link and to avoid selecting PP which is away from the MN, thus minimizing energy consumption.

We discuss our proposed ERPL in detail in the following subsections.

4.1 Objective function for selecting Preferred Parent

As of now, RPL classifies the nodes into three class namely, DAG root, FRN, and leaf node. We add another

class of node: a MN, equipped with mobility engine. The MNs are capable of changing their location and move around the network. Therefore, we need a different mechanism for the MN during the selection of the PP. Here, we discuss the OF for selecting the PP for FRN and MN. It is assumed that nodes are aware of their category during deployment over the network.

4.1.1 Objective function for fixed router node

The choice of PP among its candidate parents is done based on the new OF. The new OF is calculated using multiple routing metrics. The routing metrics used are ETX , RE , and $Dist$. The new OF is calculated as in Eq. (1):

$$OF = \alpha_1 \frac{ETX_{curr}}{ETX_{max}} 100 + \alpha_2 \left(100 - \frac{RE_{curr}}{E_{total}} 100 \right) + \alpha_3 \frac{Dist_{curr}}{Dist_{max}} 100 \quad (1)$$

where α_1 , α_2 , α_3 are weights assigned to ETX , RE , and $Dist$, respectively such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Weights can be altered according to the priority of the metrics. ETX_{curr} is link quality, ETX_{max} is maximum value an ETX can take, RE_{curr} is remaining energy of the node, E_{total} is energy of the node during deployment, $Dist_{curr}$ is distance from the node, $Dist_{max}$ is maximum distance a node can communicate. Each of the routing metrics is calculated as follows:

Expected transmission count is the expected transmissions required by the node to accomplish the task of delivering the packet to its destination successfully [24]. The value of ETX is calculated by measuring the probability that a packet reaches the neighbor (D_f) successfully and the probability that an acknowledgment packet is successfully received (D_r), as given in Eq. (2).

$$ETX_{curr} = \frac{1}{D_f \times D_r} \quad (2)$$

Residual energy is the amount of energy remaining before the node runs out of its energy. It is calculated by subtracting the amount of energy depleted from the total energy of the node, as given in Eq. (3).

$$RE_{curr} = E_{total} - E_{depletion} \quad (3)$$

where E_{total} is the initial energy of the node and $E_{depletion}$ is the energy consumed by the node. The energy depletion $E_{depletion}$ of the node is defined as the sum of energy consumed during processing, communicating, and sensing. As per [25], it is calculated as in Eq. (4).

$$E_{depletion} = V \times \left(I_{ap}T_{ap} + I_{lp}T_{lp} + I_{tx}T_{tx} + I_{rx}T_{rx} + \sum_i I_{ni}T_{ni} \right) \quad (4)$$

where V is supply voltage, $I_{ap}, I_{lp}, I_{tx}, I_{ni}$ are the current required during the microcontroller in active mode, microcontroller in low power mode, transmission mode of the communicating device, receiving mode of the communication device, sensors, respectively and $T_{ap}, T_{lp}, T_{tx}, T_{rx}, T_{ni}$ are the activity duration during the microcontroller in active mode, microcontroller in low power mode, transmission mode of the communicating device, receiving mode of the communication device, sensors, respectively.

In this work, we have used Tmote Sky provided by the Cooja simulator. Tmote Sky is an ultra-low power wireless module specifically designed for sensor networks. Figure 1 shows the structure of Tmote Sky.

Table 1 shows the current consumption of the Tmote Sky taken from the data sheet provided by the manufacturer [26, 27]. We have considered only sources which consume more energy and rest are ignored.

Distance is calculated using RSSI based on the empirical model. The advantage of using RSSI value is that it uses less costly hardware and it can be implemented in low-power devices.

Tmote Sky uses radio CC2420 which has a built-in RSSI register [26]. The register gives a digital value that can be read from the 8-bit register. The RSSI register value is continuously averaged over 8 symbol periods. The RSSI_VALID bit is set once the receiver has enabled for at least 8 symbol periods. RSSI value is computed using Eq. (5) in dBm as per [28].

$$RSSI = RSSI_VAL + RSSI_OFFSET \quad [\text{dBm}] \quad (5)$$

where the $RSSI_OFFSET$ is computed empirically during development from the front end gain. The value is set approximately to -45 [28]. The distance is calculated

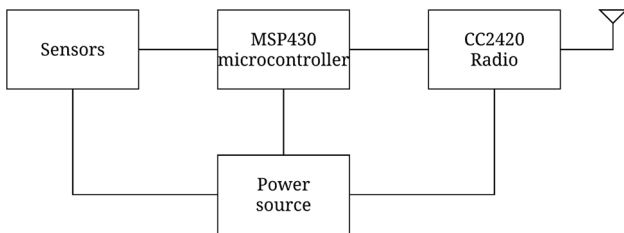


Fig. 1 Structure of Tmote Sky

Table 1 Current consumption of Tmote Sky

Module	State	Current
CPU	Active idle	600 μA 3 μA
Radio	Transmit receive	17.4 mA 19.2 mA
SHT11	Active	0.55 mA

using the relationship between RSSI value and the power of the received signal as per [29]:

$$RSSI \propto 10 \log\left(\frac{1}{D^n}\right)$$

$$RSSI = -10n \log(D) + C \quad (6)$$

where D is the distance between MN and parent node, ‘ n ’ is the path loss exponent factor, C is some fixed constant. Equation (6) can be rewritten as:

$$RSSI = -m \log(D) + C \quad (7)$$

where ‘ m ’ is the slope of linear equation between $\log(D)$ and $RSSI$. From Eqs. (6), (7) path loss exponent factor is calculated as:

$$n = \frac{m}{10} \quad (8)$$

4.1.2 Objective function for mobile node

MN has an ability to change its location in the network. Due to the mobility, if the MN is effectively distant from

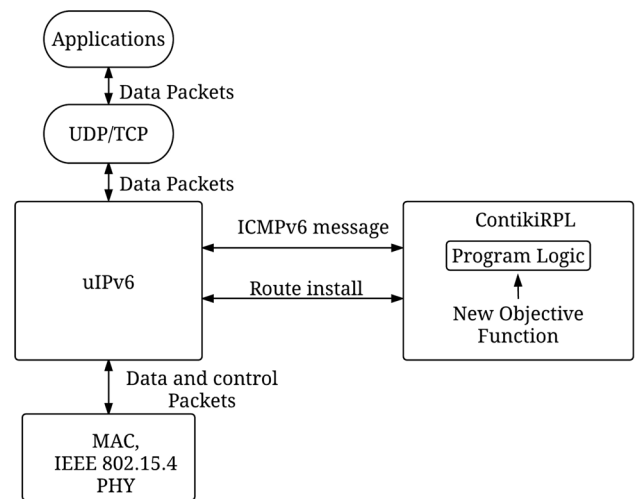


Fig. 2 Enhanced-RPL in Contiki Operating System

ID	Dist	Dist_Var	Lifetime
----	------	----------	----------

Fig. 3 Candidate_Parent_Table fields

ID	Dist	Lifetime
----	------	----------

Fig. 4 New_Parent_Table fields

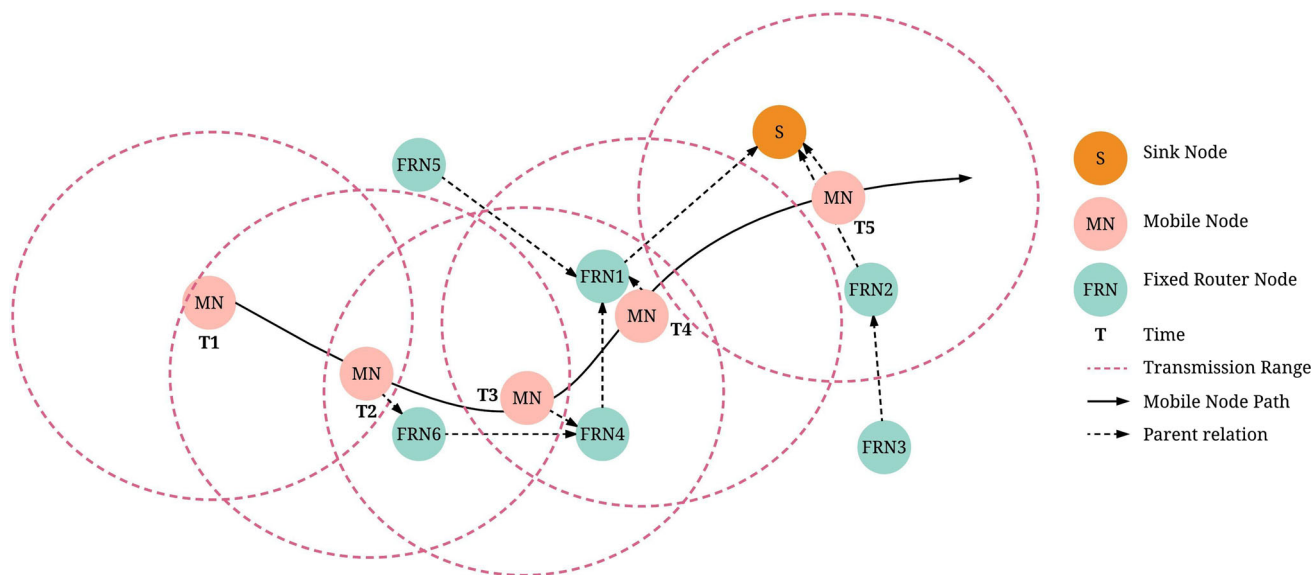


Fig. 5 Scenarios due to Mobile Node Movement

its PP then the route towards the PP becomes inconsistent. Therefore, mobility has to be given more priority over other routing metrics. A new routing metric $Dist_Var$ is defined to identify a change in position of the MN over a period of time. The new OF for the MN is calculated using multiple routing metrics i.e. ETX , $Dist$ and $Dist_Var$ as given by the Eq. (9).

$$OF = \alpha_1 \frac{ETX_{curr}}{ETX_{max}} 100 + \alpha_2 \frac{Dist_{curr}}{Dist_{max}} 100 + \alpha_3 \frac{Dist_Var}{Dist_{max}} 100 \quad (9)$$

where α_1 , α_2 , α_3 are weights assigned to ETX , $Dist$, and $Dist_Var$, respectively such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Weights can be altered according to the priority of the metrics. ETX_{curr} is link quality, ETX_{max} is maximum value an ETX can take, $Dist_{curr}$ is distance from the node, $Dist_{max}$ is maximum distance a node can communicate and this routing metric is given highest priority among other routing metrics, $Dist_Var$ is the distance variation of the node over a period of time. ETX_{curr} , $Dist_{curr}$ are calculated according to the Eqs. (2), (7), respectively whereas $Dist_Var$ is calculated using the formula given in Eq. (10).

$$Dist_Var = Dist_{t2} - Dist_{t1} \quad (10)$$

where $Dist_{t1}$, $Dist_{t2}$ are the distances from a node calculated at the time of receiving successive DIO messages. The $Dist_{t1}$, $Dist_{t2}$ are determined using $RSSI$ as per the Eq. (7). Figure 2 shows the ERPL in Contiki operating system, in which the program logic uses proposed new OF defined for mobility support in order to select the PP.

4.2 Updating Preferred Parent for Mobile Node

In case of standard RPL, the PP is updated periodically or when a link to the PP expires. However, if the MN is effectively distant from the PP i.e. MN is already out of the PP communication range then, all the packets transmitted by the MN are lost and require retransmission of the packets for successful delivery. To overcome this problem, we update the PP of the MN after receiving a DIO message from the candidate parent node or at least two DIO messages from a new parent node. In order to update the PP, the MN maintains two tables namely, *Candidate_Parent_Table* and *New_Parent_Table* and fields of these tables are given in Figs. 3 and 4, respectively. Where ID uniquely identifies the node in the network, $Dist$ indicates the distance between the parent node and MN, $Dist_Var$ designates whether the MN is moving towards or away from the parent node, $Lifetime$ indicates duration until the entry remains in the table. Algorithm 1 shows step by step procedure involved in updating the MN's PP. However, updating the PP for the FRN is same as specified in the standard RPL protocol. We discuss in detail different scenarios that arise during the selection of the PP due to the MN mobility. Figure 5 shows an example of different scenarios:

1. T1:T2 MN joining the network
2. T2:T3 MN moving close to a new parent node
3. T3:T4 MN moving close to a candidate parent node
4. T4:T5 MN moving away from the parent node

Each of these scenarios is explained in the following subsections.

Algorithm 1: Updating Preferred Parent

```

Input : Mobile Node MN
Output : PreferredParent of MN
Candidate_Parent_Table = NULL
New_Parent_Table = NULL
do
  send(Broadcast DIS message)
  if DIO == TRUE then
    Instance = DIO.Instance
    NP = New_Parent_Table.Find(Instance.ID)
    if NP == NULL then
      Dist = Calculate_Distance(RSSI)
      if Dist > Threshold then
        New_Parent_Table.Add(Instance)
      else
        Ignore DIO
      end
    else
      Compute Dist_Var
      Candidate_Parent_Table.Add(Instance)
      New_Parent_Table.Remove(Instance)
    end
  end
end
while Candidate_Parent_Table == TRUE;
Call Procedure 1
if DIO == TRUE then
  Instance = DIO.Instance
  CP = Candidate_Parent_Table.Find(Instance.ID)
  if CP == NULL then
    NP = New_Parent_Table.Find(Instance.ID)
    if NP == NULL then
      Dist = Calculate_Distance(RSSI)
      if Dist > Threshold then
        New_Parent_Table.Add(Instance)
      else
        Ignore DIO
      end
    else
      Dist = Calculate_Distance(RSSI)
      Compute Dist_Var
      Candidate_Parent_Table.Add(Instance)
      New_Parent_Table.Remove(Instance)
      Call Procedure 1
    end
  else
    if CP.ID == PreferredParent.ID then
      Dist = Calculate_Distance(RSSI)
      Compute Dist_Var
      Candidate_Parent_Table.Update(CP)
    else
      Dist = Calculate_Distance(RSSI)
      Compute Dist_Var
      Candidate_Parent_Table.Update(CP)
      Call Procedure 1
    end
  end
end
end
if PreferredParent.Lifetime == 0 then
  Candidate_Parent_Table.Remove(PreferredParent)
  PreferredParent = NULL
  Call Procedure 1
end
if CP.Lifetime == 0 then
  Candidate_Parent_Table.Remove(CP)
end
if CP == FALSE then
  repeat Algorithm 1
end

```

Procedure 1: Preferred Parent Selection

```

PreferredParent = NULL
MAX_OF = Maximum_Value
if Candidate_Parent_Table == TRUE then
  foreach CP ∈ Candidate_Parent_Table do
    if CP.OF < MAX_OF then
      MAX_OF = CP.OF
      PreferredParent = CP
    end
  end
end
if PreferredParent then
  Send(DAO message to PreferredParent)
else
  repeat Algorithm 1
end

```

4.2.1 Mobile Node joining the network

In this scenario, we discuss about steps involved during the MN joining the network. Initially, MN's *Candidate_Parent_Table* and *New_Parent_Table* are empty. Since, no parent nodes are available in the *Candidate_Parent_Table*, MN keeps broadcast DIS message into the network until a parent node is available in the *Candidate_Parent_Table*. If it receives a DIO message from the parent nodes, it calculates the distance based on the *RSSI* value. If the measured distance is above the *threshold* then the parent node is added to *New_Parent_Table* with all the information. If a node receives another DIO message from the same parent node within the *Lifetime*, then *Dist_Var* is computed. Then, the parent node information is added to *Candidate_Parent_Table* by removing the entry in the *New_Parent_Table*. If this is the first candidate parent node then it is selected as PP otherwise, candidate parent node with least OF is selected as PP.

In Fig. 5 at T1, MN keeps broadcast DIS message since there are no candidate parents, at T2, FRN6 is the only candidate parent node, therefore, it is selected as PP node.

4.2.2 Mobile Node moving close to a new parent

In this scenario, we discuss the steps that take place when a MN moving close to a new parent node. When MN receives a DIO message from a parent node, it checks if the parent node is already present in *Candidate_Parent_Table*. If the parent node is not present, then it searches in *New_Parent_Table*. If the parent node is already present, then, it determines *Dist* from *RSSI* of the DIO message, computes *Dist_Var*, add the parent node to *Candidate_Parent_Table*, removes its entry from the *New_Parent_Table* and checks if the parent node has potential to become PP. Otherwise, it determines *Dist* from *RSSI* of the DIO message and compares with the *threshold*. If *Dist* is

greater than the *threshold*, then it is added to the *New_Parent_Table* otherwise DIO message is ignored.

In Fig. 5 at T3 since the MN moved close to FRN4, therefore has least OF as compared to FRN6 and FRN1, FRN4 is selected as PP.

4.2.3 Mobile Node moving close to a candidate parent

In this scenario, we discuss MN moving close to a candidate parent node. When the MN receives a DIO message from the candidate parent node, it checks if the candidate parent node is already a PP or not. If the candidate parent node is already the PP then, *Dist* is determined using *RSSI* of the DIO message and *Dist_Var* is updated in the *Candidate_Parent_Table* otherwise, the corresponding parent node is updated and the candidate parent node with least OF is selected as PP.

In Fig. 5 at T3, FRN4 is selected as the PP and FRN1 is already a candidate parent node of the MN. When a MN moves close to the FRN1 at T4, FRN1 has least OF, hence selected as PP.

4.2.4 Mobile Node moving away from the parent

In this scenario, we discuss MN moving away from a parent node i.e. either candidate parent node or new parent. When the MN receives a DIO message from the parent node, it adds a timer or reset the timer in the corresponding *Candidate_Parent_Table* or *New_Parent_Table*. If the MN does not receive a DIO message before the *Lifetime* of the entry in the *Candidate_Parent_Table* or *New_Parent_Table* expires then it is assumed that MN is moved away from that parent node and no longer reachable, parent node entry in the corresponding table is removed. If the removed parent node is PP then a new candidate parent node is selected as a PP with least OF.

In Fig. 5 at T4, FRN1 is selected as PP. At T5, MN moved away from the FRN1, it does not hear any DIO message further from it. Therefore *Lifetime* expires and is removed from the *Candidate_Parent_Table*.

5 Simulation results and discussion

In order to evaluate the proposed ERPL, we have used Cooja simulator provided by the Instant Contiki 2.6 [30, 31], and the parameters used in our work are shown in Table 2. To generate the mobility traces, we have used the bonnmotion software [32], the parameters are shown in the Table 3. The performance of the ERPL is compared with standard RPL (SRPL) [6] and MRPL [12] which calculates the DIS interval using distance and speed of the MN.

Table 2 Simulation parameters

Simulation parameter	Values
Radio model	Unit disk graph model
Transmission range	30 m
Mote type	Sky Mote
Network size	100 m × 100 m
Node density	FRN: 30, MN: 9
Mobility model	Random waypoint
Simulation time	600 s

Table 3 Bonnmotion software parameters

Settings	Area	Speed	Nodes
Value	10,000 m ²	(0.5, 1, 1.5, 2) m/s	9

Simulation results are obtained by varying the speed of the MN, the number of packet transmission, and trickle timer under the grid and random topologies. In grid topology, FRNs are placed at a certain distance from each other, whereas in case of random topology, FRNs are placed randomly such that at least one FRN is in the communication range of other FRN, similar to most actual environments.

Reason for varying speed The main goal of this work is to update neighbors whenever MN changes its position. Therefore, to analyze the performance of the ERPL, MN speed is set to 0.5, 1, 1.5 and 2 m/s, which is in the range of human walking and running speed.

Reason for varying traffic Packets get lost when the MN moves away from the communication range of the PP and require retransmission of packets to deliver successfully to the destination. To analyze the performance of the ERPL, packet transmission rate is set to 1 packet per 2 minutes, 1 packet per minute, 2 packets per minute, and 3 packets per minute.

Reason for varying trickle timer Since the PP is updated based on the OF which is obtained from the DIO message which in turn broadcast based on trickle timer. Hence, trickle timer is varied to analyze the performance of the ERPL. At first, the timer is fixed to I_{min} value 12, then $[I_{min}, I_{max}]$ is set to $[10, 8]$, $[12, 8]$ and $[14, 8]$.

To measure the performance of the ERPL, we have considered power consumption, packet overhead, PDR and latency as evaluation metrics. Importance of these metrics and the results obtained from the simulations are explained below.

5.1 Analysis of power consumption

Energy consumption is an important factor for a successful sensor network operation since the sensor nodes are battery powered. To calculate power consumption, we have used power trace tool [33] available in the Cooja simulator. It uses power state monitoring to calculate the system power consumption.

Figure 6 shows the power consumption in the grid and random topologies which are sub-categorized with respect to varying speed, trickle timer, and traffic.

From Fig. 6a, it is found that proposed ERPL consumes on an average 17% less power as compared to SRPL and on an average 10% less power as compared to MRPL in the grid topology. In random topology, as shown in Fig. 6d, it is observed that ERPL consumes on an average 12% less power as compared to SRPL and on an average 4% less as compared to MRPL by varying speed of the MNs.

From Fig. 6b, it is found that ERPL outperforms with respect to SRPL by 17% on an average and MRPL by 3% on an average in the grid topology. In random topology, ERPL outperforms on an average 15% as compared to SRPL and by 3% as compared to MRPL by varying trickle timer, as shown in Fig. 6e.

As shown in Fig. 6c, the average power consumption of ERPL is 17% less as compared to SRPL and 10% less as compared to MRPL in the grid topology. In random topology, as shown in Fig. 6f, the proposed ERPL consumes on an average 21% less power as compared to SRPL

and 9% less power as compared to MRPL by varying traffic.

The main reason for SRPL to consume more power is that there is a chance of MN becoming PP, which may not be available after a certain period due to its movement. Therefore, SRPL requires more control messages and retransmission of packets which results in consuming more power. In case of MRPL, even though MN is moving away and distant from the PP, still packets are forwarded to it which may result in packet loss. Hence, MRPL consumes more power. Another reason is that MRPL requires control messages to find a new PP when the MN moves away from the PP. By contrast, ERPL selects based on the direction of the MN movement. Therefore, FRN which is very close to the MN is selected as PP. Hence, ERPL showed improvement in all the situations.

5.2 Analysis of packet overhead

Control packets are essential for constructing, sustaining and repairing routes in the network. Hence, considered them as an important component of the network. However, redundant transmission of the control packets consumes more energy, overload the network, and influence on the transmission of other packets. Therefore, RPL should contain the transmission of control messages. Packet overhead is calculated as the total number of control messages i.e., DIS, DIO, DAO, DAO-ACK transmitted until completion of the simulation.

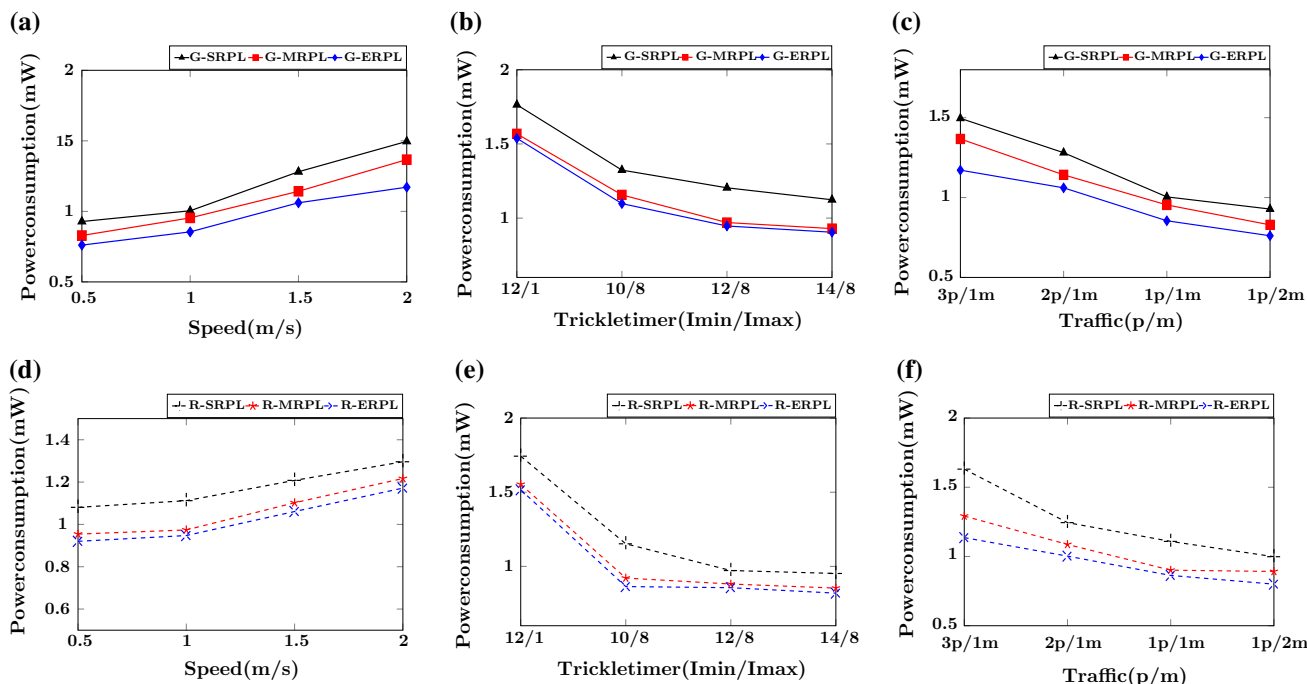


Fig. 6 Power consumption in grid (G) [random (R)] topology with respect to Speed, Trickle timer, Traffic: a–c [d–f]

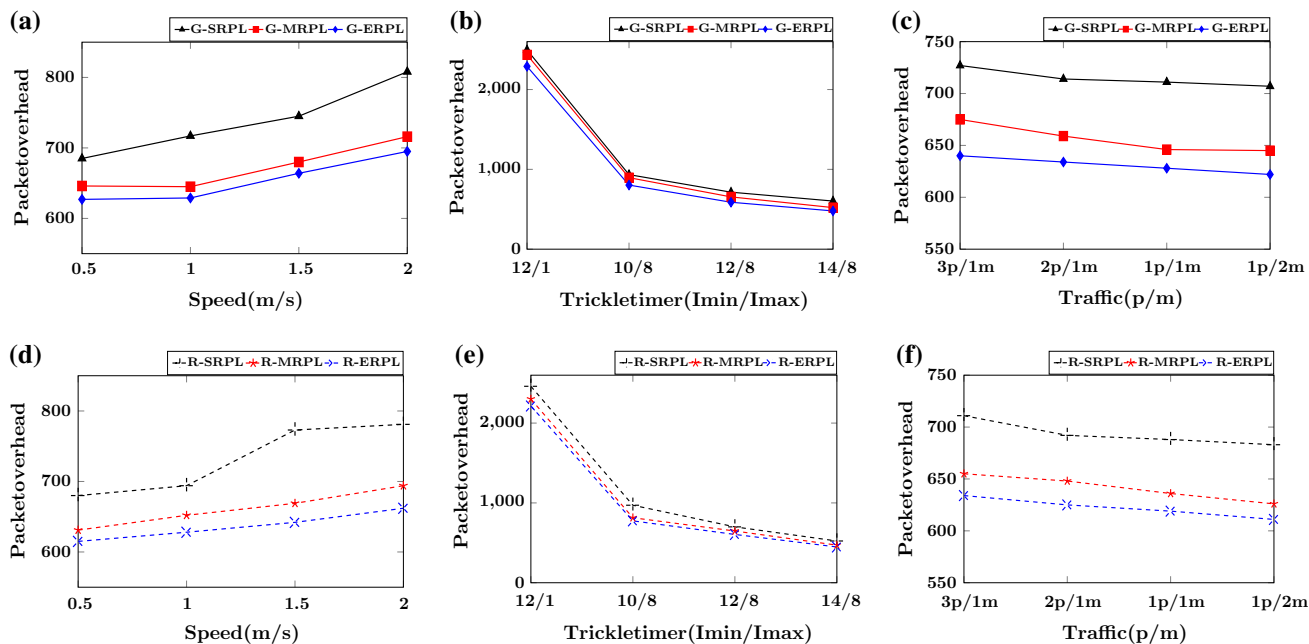


Fig. 7 Packet overhead in grid (G) [random (R)] topology with respect to Speed, Trickle timer, Traffic: **a–c** [**d–f**]

Figure 7 shows the packet overhead in the grid and random topologies which are sub-categorized with respect to varying speed, trickle timer, and traffic.

Figure 7a, d show the impact of varying speed of the MNs on a number of control messages transmitted over the network in the grid and random topologies, respectively. The ERPL requires on an average 11% fewer control messages as compared to SRPL and 2% fewer control messages than MRPL in the grid topology. In random topology, ERPL requires on an average 12% fewer control messages as compared to SRPL and 3% fewer control messages than MRPL.

Figure 7b, e show the impact of varying trickle timer on a number of control messages transmitted over the network in the grid and random topologies, respectively. The ERPL requires on an average 15% fewer control messages as compared to SRPL and 8% fewer control messages than MRPL in the grid topology. In random topology, ERPL requires on an average 14% fewer control messages as compared to SRPL and 5% fewer control messages than MRPL.

Figure 7c, f show the impact of varying traffic on a number of control messages transmitted over the network in the grid and random topologies, respectively. The ERPL requires on an average 11% fewer control messages as compared to SRPL and 3% fewer control messages than MRPL in the grid topology. In random topology, ERPL requires on an average 10% fewer control messages as compared to SRPL and 2% fewer control messages than MRPL.

The reason for overhead in SRPL is that MNs are allowed to broadcast DIO messages, which significantly increases the number of control messages in the network. If the MN becomes a PP for some node then due to the mobility, it is not available for the node after a certain period of time. To repair the path, the node sends control messages which further increases the number of control messages in the network. In MRPL, when the MN moves away from the PP, the node solicits DIS message in the network to which neighboring nodes reply with DIO message. This procedure is repeated to update the PP, which increases control messages in the network. By contrast, in ERPL the MNs are not allowed to send DIO messages. Therefore not selected as PP at any movement. For updating the PP, the FRN which is close to the MN is selected as PP. Therefore, requires fewer control messages in all situations.

5.3 Analysis of packet delivery ratio

The basic function of the network is to transfer and receive packets from source to the destination. Therefore, PDR is an essential parameter to analyze the network performance. Normally, a network with high PDR is considered as effective and applicable. PDR is calculated as the total number of packets that successfully reached the sink node to the sum of packets sent by all nodes in the network.

Figure 8 shows the PDR under the grid and random topologies which are sub-categorized with respect to varying speed, trickle timer, and traffic.

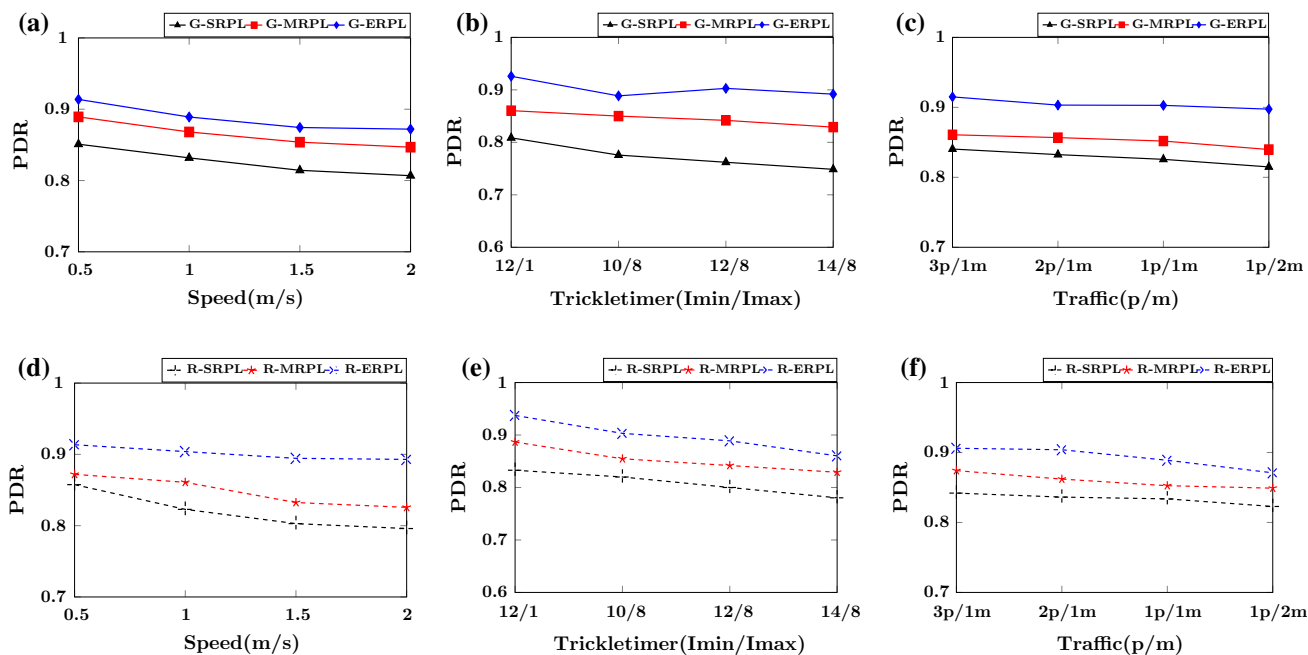


Fig. 8 PDR in grid (G) [random (R)] topology with respect to Speed, Trickle timer, Traffic: **a–c** [d–f]

Figure 8a, d show the influence of varying the speed on PDR over the network in the grid and random topologies, respectively. The average PDR in the grid topology was found to be 8% improvement for the ERPL as compared to SRPL and 3% improvement for the ERPL against MRPL. In random topology, it is found that ERPL showed 10% better PDR against SRPL and 6% better against MRPL.

Figure 8b, e show the impact of varying trickle timer on PDR over the network in the grid and random topologies, respectively. The average PDR in the grid topology is found to be 16% improvement for the ERPL as compared to SRPL and 6% improvement in the ERPL against MRPL. In random topology, it is found that ERPL showed 11% better PDR against SRPL and 5% better against MRPL.

Figure 8c, f show the impact of varying traffic on PDR over the network in the grid and random topologies, respectively. The average PDR in the grid topology found to be 9% improvement for the ERPL as compared to SRPL and 6% improvement in the ERPL against MRPL. In random topology, it is found that ERPL showed 7% better PDR against SRPL and 3% better against MRPL.

The reason for low PDR in SRPL is that all the nodes are treated equally irrespective of their mobility status. PDR decreases drastically when the MN is selected as PP since the MN changes their location, new PP is not updated quickly. In MRPL, the MN solicits DIS message when leaving time of MN expires. To select new PP, MRPL takes some amount of time which results in packet loss. Another reason for low PDR is that even though MN is

moving away and distant from the PP, still packets are forwarded to it which results in packet loss. By contrast in ERPL, a new PP is selected when the MN moves close to another FRN without waiting for route expiration time. Hence, showed better results in all the situations.

5.4 Analysis of latency

This metric ought to be limited for applications that require real-time guarantees. It is calculated as the time taken by the packets from the point of transmission to its reception at the sink node.

Figure 9 shows the latency under the grid and random topologies which is sub-categorized with respect to varying speed, trickle timer, and traffic.

From Fig. 9a, it is found that ERPL packets reach the destination faster than SRPL by an average 8% and an average 5% as compared to MRPL in grid topology. In random topology, as shown in Fig. 9d, it is found that ERPL packets reach the destination faster by an average 9% as compared to SRPL and an average 5% as compared to MRPL by varying speed of the MNs.

From Fig. 9b, it is found that ERPL packets reach the destination faster than SRPL by an average 5% and an average 2% compared to MRPL in grid topology. In random topology, as shown in Fig. 9e, it is found that ERPL packets reach the destination faster by an average 7% as compared to SRPL and an average 2% as compared to MRPL by varying trickle timer.

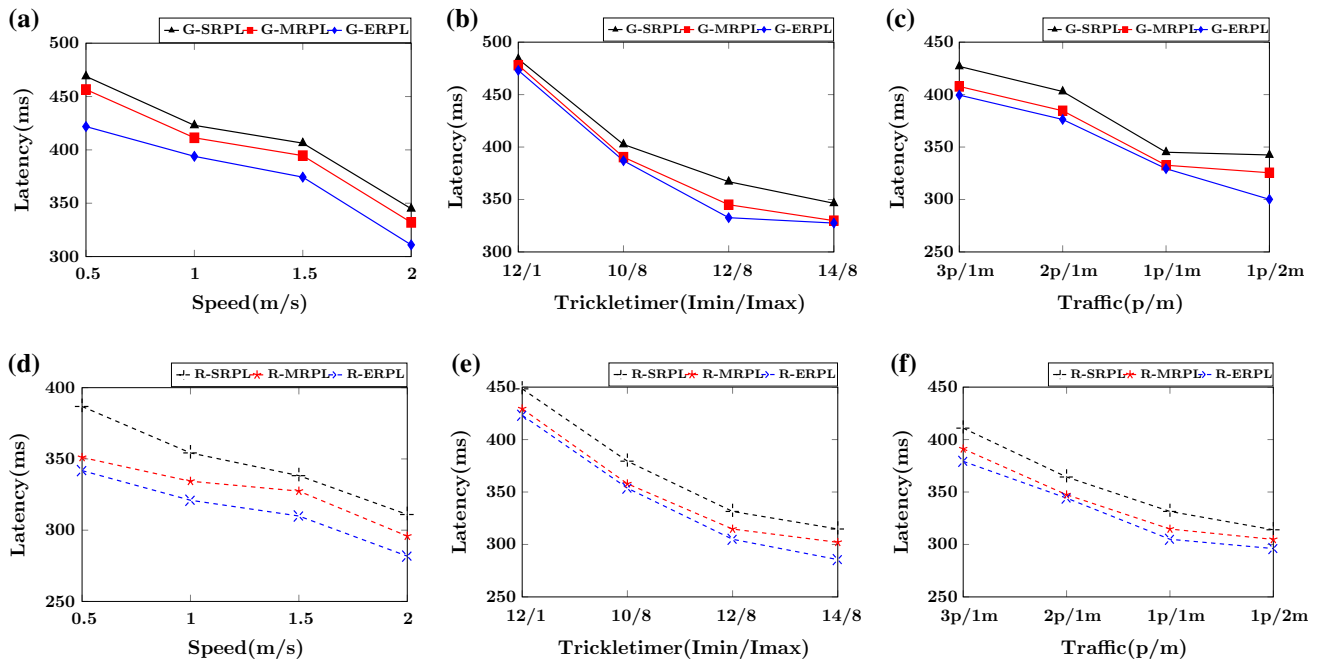


Fig. 9 Latency in grid (G) [random (R)] topology with respect to Speed, Trickle timer, Traffic: **a–c** [d–f]

As shown in Fig. 9c, it is found that ERPL packets reach the destination faster than SRPL by an average 7% and on an average 3% as compared to MRPL in grid topology. In random topology, as shown in Fig. 9f, it is found that ERPL packets reach the destination faster by an average 6% as compared to SRPL and on an average 2% as compared to MRPL by varying traffic.

In case of SRPL, if the intermediate nodes along the path from source node to destination node are Mobile Nodes, then packets take longer time to reach the destination. In case of MRPL, as the distance between the node and its PP increases latency also increases. By contrast, ERPL makes use of distance as a routing metric while calculating OF. Hence, showed a slight improvement in all the situations.

5.5 Evaluating the impact of fixed router node density

In this study, we analyze the performance of ERPL by varying the density of FRNs by 15, 20, 25 and 30 nodes in the random topology. In general, the proposed ERPL performance increases with increase in density of FRNs, as shown in Fig. 10.

Initially, with 15 FRNs the proposed ERPL achieved approximately 81% of PDR. As the number of FRNs is increased to 30 nodes, ERPL achieved approximately 89% of PDR. In general, as the density of FRNs

increases the MNs will have more candidate parents that increase the probability of selecting the best parent with the high-quality link, therefore, increases the PDR. However, with the low density of FRNs, the ERPL performs better as compared to SRPL and MRPL, as shown in Fig. 10a.

Figure 10b shows the average power consumption of the nodes in the network. The average power consumption of ERPL decreases with increase in FRNs. The power consumption of node includes the power spent over the transmission of control messages i.e. DIS, DIO, DAO, and DAO-ACK messages. With less number of FRNs, node requires a more control messages for selecting a PP, therefore, consume more power. However, ERPL consumes less power as compared to SRPL and MRPL.

Figure 10c shows that average delay of ERPL reduces slightly with an increase in the number of FRNs. With less number of FRNs, there is a chance of selecting PP which is far from the node. In case of ERPL with increasing number of FRNs, there is a chance of selecting PP which is near to the node. However, in SRPL node selection is based only on ETX value which increases the delay.

Figure 10d shows the impact of varying the density of FRNs on a number of control messages transmitted over the network. The total number of control messages increase with increase in FRNs as each node sends control messages to construct and maintain DODAG. However, the number of control messages required for selecting a PP decreases with increase in FRNs since the MN rarely broadcast DIS

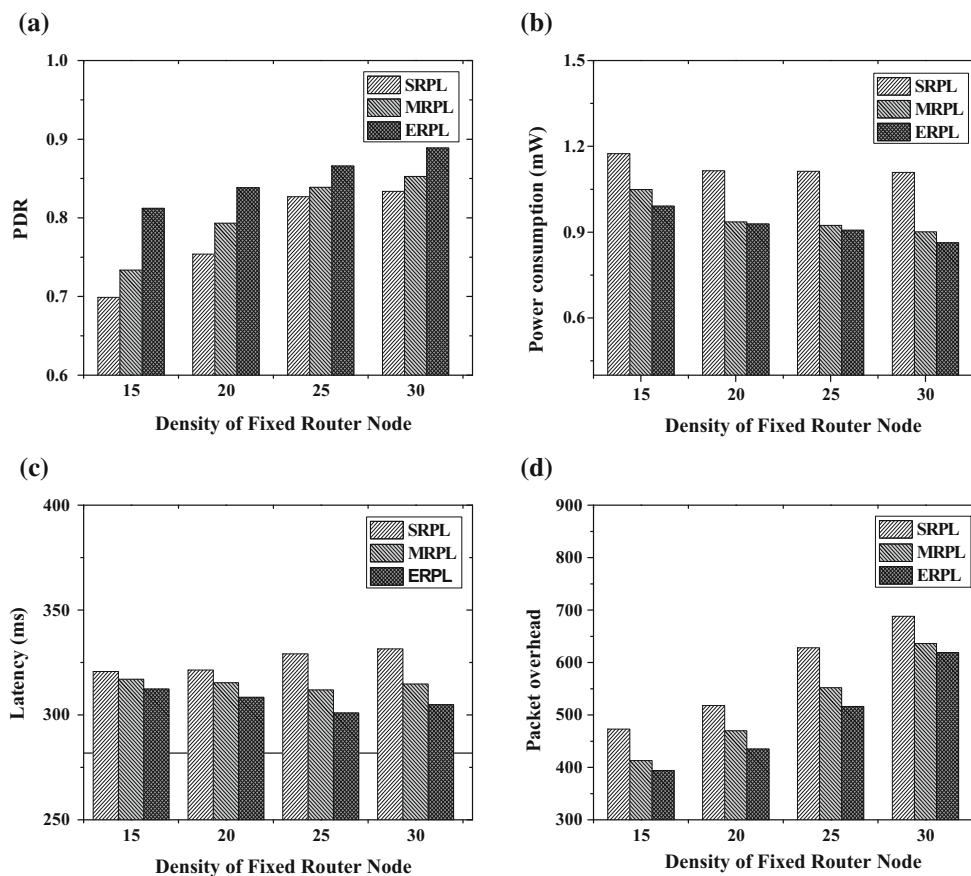


Fig. 10 Impact of Fixed Router Node density on **a**, **b** power consumption, **c** latency, and **d** packet overhead

message as there is more chance of finding candidate parent with high density of FRNs.

6 Discussions

From the simulation results, we observed that the ERPL performs better by updating PP quickly whenever a MN moves away from already selected PP under different traffic and system parameters. Nevertheless, there are several considerations regarding the viability of the ERPL.

- A node must be able to recognize whether it is static or mobile in nature, as MN is not allowed to transmit DIO message.
- The PP is selected based on OF that includes distance as a routing metric. Distance is calculated using *RSSI* value which is sensitive to interferences in the real world scenario.
- The ERPL requires MNs to be in the range of at least one FRN since MNs are not allowed to become PP.
- To achieve high performance, the ERPL requires a more number of FRNs in the network.

- Finally, the power consumption and packet overhead can be reduced further if ERPL predicts the path of the MN and selects an ideal FRN as PP.

7 Conclusion

An ERPL routing protocol is proposed in this work to provide mobility support for LLNs. In addition, we have introduced a new OF that selects PP based on high quality, high availability and low power consumption. On the basis of the above, we also proposed a mechanism to update PP for MN. The proposed ERPL has been simulated using Cooja simulator in order to verify its effectiveness. Our empirical results showed that proposed ERPL outperformed MRPL and SRPL by achieving the maximum of 21% less power consumption, 15% of less packet overhead, 9% of low latency and PDR increased to 16%. The future aim is to explore new routing metrics that support mobility and analyze the performance of ERPL along with different mobility models.

References

- Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15), 2787–2805.
- urii, M. P., Tafa, Z., Dimi, G., & Milutinovi, V. (2012). A survey of military applications of wireless sensor networks. In *2012 Mediterranean conference on embedded computing (MECO)*, pp 196–199.
- Othman, M. F., & Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. *Procedia engineering*. In *International symposium on robotics and intelligent sensors*, vol. 41, pp. 1204–1210. (IRIS 2012).
- Alemdar, H., & Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15), 2688–2710.
- Rehan, W., Fischer, S., Rehan, M., & Rehmani, M. H. (2017). A comprehensive survey on multichannel routing in wireless sensor networks. *Journal of Network and Computer Applications*, 95, 1–25.
- Winter, T. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks.
- Ko, J., Terzis, A., Dawson-Haggerty, S., Culler, D. E., Hui, J. W., & Levis, P. (2011). Connecting low-power and lossy networks to the internet. *IEEE Communications Magazine*, 49(4), 96–101.
- Gnawali, O. (2012). The minimum rank with hysteresis objective function.
- Carels, D., Poorter, E. D., Moerman, I., & Demeester, P. (2015). Rpl mobility support for point-to-point traffic flows towards mobile nodes. *International Journal of Distributed Sensor Networks*, 11(6), 470349.
- Korbi, I. E., Brahim, M. B., Adjih, C., & Saidane, L. A. (2012). Mobility enhanced rpl for wireless sensor networks. In *2012 Third international conference on the network of the future (NOF)*, pp. 1–8.
- Cobran, C., Montavont, J., & Nol, T. (2014). Analysis and performance evaluation of rpl under mobility. In *2014 IEEE symposium on computers and communications (ISCC)*, pp. 1–6.
- Gara, F., Saad, L. B., Hamida, E. B., Tourancheau, B., & Ayed, R. B. (2016). An adaptive timer for rpl to handle mobility in wireless sensor networks. In *2016 International wireless communications and mobile computing conference (IWCMC)*, pp. 678–683.
- Park, J., Kim, K. H., & Kim, K. (2017). An algorithm for timely transmission of solicitation messages in rpl for energy-efficient node mobility. *Sensors*, 17(4), 899.
- Levis, P., Patel, N., Culler, D., & Shenker, S. (2004). Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the 1st conference on symposium on networked systems design and implementation—volume 1*, USENIX Association, Berkeley, CA, USA, NSDI'04, pp. 2–2.
- Khan, A. A., Rehmani, M. H., & Reisslein, M. (2017). Requirements, design challenges, and review of routing and mac protocols for cr-based smart grid systems. *IEEE Communications Magazine*, 55(5), 206–215.
- Ko, J., & Chang, M. (2015). Momoro: Providing mobility support for low-power wireless applications. *IEEE Systems Journal*, 9(2), 585–594.
- Lee, K. C., Sudhaakar, R., Dai, L., Addepalli, S., & Gerla, M. (2012). Rpl under mobility. In *2012 IEEE consumer communications and networking conference (CCNC)*, pp. 300–304.
- Cobârzan, C., Montavont, J., & Noel, T. (2015). Integrating mobility in rpl. In *European conference on wireless sensor networks*, Springer, pp. 135–150.
- Kuntz, R., Montavont, J., & Noël, T. (2013). Improving the medium access in highly mobile wireless sensor networks. *Telecommunication Systems*, 52(4), 2437–2458.
- Barcelo, M., Correa, A., Vicario, J. L., Morell, A., & Vilajosana, X. (2016). Addressing mobility in rpl with position assisted metrics. *IEEE Sensors Journal*, 16(7), 2151–2161.
- Gaddour, O., Kouba, A., Rangarajan, R., Cheikhrouhou, O., Tovar, E., & Abid, M. (2014). Co-rpl: Rpl routing for mobile low power wireless sensor networks using corona mechanism. In *Proceedings of the 9th IEEE international symposium on industrial embedded systems (SIES 2014)*, pp. 200–209.
- Fotouhi, H., Moreira, D., & Alves, M. (2014). *Mobile iot: smart-hop over rpl*. CISTER-Research Centre in Realtime and Embedded Computing Systems: Tech. rep.
- Saleem, Y., Yau, K. L. A., Mohamad, H., Ramli, N., Rehmani, M. H., & Ni, Q. (2017). Clustering and reinforcement-learning-based routing for cognitive radio networks. *IEEE Wireless Communications*, 24(4), 146–151.
- Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on mobile computing and networking*, ACM, New York, NY, USA, MobiCom '04, pp. 114–128.
- Dunkels, A., Osterlind, F., Tsiftes, N., & He, Z. (2007). Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th workshop on embedded networked sensors*, ACM, New York, NY, USA, EmNets '07, pp. 28–32.
- Sky, T. (2006). Ultra low power ieee 802.15. 4 compliant wireless sensor module. Moteiv Corporation
- Amiri, M. (2010). Measurements of energy consumption and execution time of different operations on tmote sky sensor nodes. Ph.D. thesis, Masarykova univerzita, Fakulta informatiky.
- Instruments, T. (2007). Cc2420 datasheet. Reference SWRS041B.
- Kumar, P., Reddy, L., & Varma, S. (2009). Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks. In *2009 Fifth international conference on wireless communication and sensor networks (WCSN)*, pp. 1–4.
- Jevtić, M., Zogović, N., & Dimić, G. (2009). Evaluation of wireless sensor network simulators. In *Proceedings of the 17th telecommunications forum (TELFOR 2009)*, Belgrade, Serbia, pp. 1303–1306.
- Dunkels, A. (2012). *Contiki operating system*. <http://www.contiki-os.org/>.
- Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., & Schwamborn, M. (2010). Bonnmotion: A mobility scenario generation and analysis tool. In *Proceedings of the 3rd international ICST conference on simulation tools and techniques*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), p. 51.
- Dunkels, A., Eriksson, J., Finne, N., & Tsiftes, N. (2011). Powertrace: Network-level power profiling for low-power wireless networks.



Shridhar Sanshi received M.Tech. degree in Information Technology from National Institute of Technology Karnataka, Surathkal, Karnataka, India, in 2013. He is currently pursuing Ph.D. in Information Technology at National Institute of Technology Karnataka, Surathkal, Karnataka, India. His research areas of interest include Wireless Sensor Network, Ad-hoc Networks and Internet of Things.



C. D. Jaidhar received Ph.D. degree in computer science and engineering from NIT Trichy Tamilnadu, in 2008. He is currently with Department of Information Technology, National Institute of Technology, Karnataka, Surathkal, India, where he is working an Assistant Professor, since 2013. His research interests include computer networks, cryptography and network security, malware analysis, machine learning, ad-hoc networks,

wireless sensor network and Internet of Things.