An Adaptive Timer for RPL to handle Mobility in Wireless Sensor Networks

Fatma Gara†, Leila Ben Saad‡, Elyes Ben Hamida†, Bernard Tourancheau†, Rahma Ben Ayed‡

† University of El Manar, ENIT, BP 37 LE BELVEDERE 1002, Tunisia
‡ Department of Information & Communication Technology, University of Agder, Norway
§ QMIC, Qatar Science and Technology Park PO Box 210531, Doha, Qatar
† LIG, University of Grenoble, F-38402 Saint Martin d’Heres, France

fatmagara@gmail.com, leila.bensaad@ui.no, elyesb@qmic.com
bernard.tourancheau@imag.fr, rahma.benayed@enit.rnu.tn

Abstract—This paper focuses on the performance of wireless sensor networks characterized by a hybrid topology composed of mobile and static sensor nodes. The Routing Protocol for Low power and lossy networks (RPL), which is standardized as an IPv6 routing protocol for low power and lossy networks, uses the trickle timer algorithm to handle changes in the network topology. However, this algorithm is not well adapted to dynamic environments. This paper enhances the trickle timer in order to fit with mobility requirements. Most of previous works have improved this algorithm without considering the random movement of nodes. In this work, the proposed timer algorithm takes into consideration the random trajectory of mobile nodes, pause time and node’s velocity. It is also dynamically adjusted to prevent from node disconnections. The performance of the modified protocol is evaluated and compared with native RPL, MERPL and RPL with reverse trickle. The results show that our protocol optimization offers better performance.

Index Terms—Wireless sensor networks; RPL protocol; mobility; trickle timer; adaptive timer.

I. INTRODUCTION

Wireless sensor networks (WSNs) are considered among emerging technologies through a new popular paradigm in research and industry, which is the Internet of Things (IoT) [1]. With this concept, billions of sensors will be deployed in the world. However, the hypothesis of fixed nodes could no longer be considered because recent trends show that the mobility has become a requirement for certain types of applications such as healthcare applications [2] and intelligent transport systems [3]. In these applications, sensors which are placed on patients or vehicles, are mobile and need to transmit the collected data quickly and reliably to the sink.

In this context of connecting small objects to the Internet, the ROLL Working Group of Internet Engineering Task Force (IETF) has standardized an IPv6 Routing Protocol for Low-power and lossy networks (RPL) [4]. Although this protocol is designed for static networks, it can be adapted to a hybrid network topology characterized by a continuous change due to the mobility of some sensors. This protocol uses the trickle timer algorithm to handle changes in the network topology by adjusting intervals of sending control messages. However, this algorithm is conceived basically for static networks. Many works presented in literature have focused on adapting the trickle timer to mobility. Nevertheless, to our knowledge, no work has considered the random aspect of node’s mobility. Their assumptions are mainly based on historical node mobility to predict the future movement or node stability that decreases with time.

In this paper, we propose a new adaptive timer algorithm which takes into consideration the random trajectory of mobile nodes. This algorithm is dynamically adjusted based on the time to leave in order to prevent from mobile node disconnections. Moreover, the RPL protocol is improved by changing the metric used to elect the preferred parent.

The remainder of this paper is organized as follows. Section 2 reviews the main studies conducted to improve the performance of RPL protocol in the context of mobility. In Section 3, an overview of the RPL protocol is presented. Section 4 studies how to modify RPL protocol to fit with the needs of mobility requirements. Section 6 evaluates the performance of the modified RPL and compares it with native RPL, MERPL and RPL with reverse trickle. Section 7 concludes the paper.

II. RELATED WORK

In this section, an overview of existing works that focus on handling mobility in WSNs is presented. These works are based on the modification of the RPL protocol because it has become a standard since 2012 [4].

In [5], the authors propose mRPL which integrates Smart-Hop algorithm into RPL. The Smart-Hop algorithm has two phases: Data Transmission Phase and Discovery Phase. In Data Transmission Phase, the mobile node is assumed to have a reliable link with an Access Point (AP). The mobile node monitors the link quality by receiving reply packets from the serving AP. Upon receiving n data packets in a given window, the serving AP replies with the average RSSI of the n packets. The smart hop algorithm schedules, inside the Destination Oriented Directed Acyclic Graph (DODAG), the exchanged control messages i.e. DODAG Information Solicitation (DIS), Destination Advertisement Object (DAO) and DODAG Information Object (DIO) by using the trickle timer algorithm. A mobile node chooses as a preferred parent a node that does not belong to its sub-DODAG and has the highest average Received Signal Strength Indicator (RSSI).
The authors also implemented timers in order to improve the effectiveness of the Hand-off procedure. The simulation results showed that the mRPL algorithm minimizes the overhead and improves the packet delivery ratio. However, the authors did not use random topologies in their simulations. Moreover, a mobile node always chooses as a preferred parent an access point. This means that the mobile node never selects a mobile node as a preferred parent which may lead to disconnections in a random topology. In addition to that, the choice of the preferred parent is mainly based on the RSSI value which is very sensitive to interferences.

In [6], the authors proposed an additional layer to handle mobile nodes detection. The MoMoRo algorithm collects information from neighbors by broadcasting beacon messages and collecting replies. The authors also used fuzzy estimators to estimate link quality. Moreover, MoMoRo allows disconnected mobile nodes to quickly reconnect if there are packets in the network destined to them. However, in this scheme, the mobility detection is only based on packet loss. Since WSNs are characterized by lossy links, this approach increases the packet loss ratio and leads to an important overhead.

In [7], the authors propose a new scheme for RPL protocol called MERPL to handle mobility in WSNs. First, they identify mobile nodes by introducing a new flag in DIS messages. Second, the choice of the preferred parent is modified and it is determined not based on computing an objective function but on the identification of nodes. Fixed nodes are usually chosen in priority. But if nodes are similar, the objective function is used. This scheme increases the reliability but it may increase the latency. In addition to that, the authors regulate the transmission of DIS messages which are used by nodes to join the network and to refresh their attachment to the preferred parent. If a mobile node chooses many preferred parents, it can disconnect at any time and the interval of DIS messages will be minimized. But, if a mobile node was stable in the past, it means according to the authors that it would be stable in the future. So, the DIS interval will be maximized. However, this procedure of adapting the DIS interval based on the history of the mobile node does not necessary detect rapidly the disconnection of the mobile node and prevent the packet loss.

The authors in [8] introduced extensions to RPL in order to efficiently support mobile nodes. In fact, they proposed a reverse trickle timer algorithm that allows mobile nodes to quickly detect the unreachability of their current parents and re-attach to the graph. Whenever a mobile node sends to his preferred parent a DAO message with a mobility flag set to 1, the preferred parent will use the reverse trickle timer. The transmission interval of DAO messages of preferred parents that have in their sub-DODAG mobile nodes should be at first large and then it decreases. In fact according to the authors, mobile nodes’ stability decreases with time. In addition to that, the preferred parent will multicast DIO with a Destination Advertisement Trigger Sequence Number (DTSN) that triggers the sending of a DAO from any attached node. If all DAO messages are received with a mobility flag set to 0, the parent switches to the trickle timer algorithm. The proposed protocol improves the packet delivery ratio. However, the assumption that the stability of mobile nodes decreases with time is not always true. In fact, the mobile node can be initially in motion. Then, it may stop for a long time which means that its stability increases.

In [9], the authors propose CO-RPL protocol which introduces modifications to RPL protocol to handle mobility in wireless sensor networks. The basic idea is to divide the network into circular area (i.e. corona). Each node belongs to one corona and has a unique identifier (i.e. corona id). The choice of the preferred parent is based on the rank and the corona id of each node. This corona mechanism aims to prevent loops in the network. The authors also modified the trickle timer. The DIO message interval is fixed and it is adjusted only at the beginning based on the speed of the nodes. This protocol prevents packet loss by fast detecting the disconnection of mobile node. But in the other hand, it drastically increases the control message overhead and power consumption of the nodes.

In our previous work [10], an enhancement of RPL protocol to support short mobility of some sensors like in healthcare applications is proposed. The modified protocol proposes to make mobile sensors leaf nodes in the DODAG. This is achieved by preventing mobile nodes from sending DIO messages. The simulation results show that the protocol offers better performance than native RPL in terms of overhead, power consumption, packet delivery ratio and delay. This paper extends our previous work by adapting RPL protocol to mobility in random wireless sensor network topology with a new adaptive timer algorithm.

III. RPL protocol overview

RPL has become a standard since 2012 by the IETF [4]. It mitigates the problems created by other protocols for Low power and Lossy networks. The basic idea of RPL is that each node has multiple paths toward the root node. But, it has a preferred one through its preferred parent. The network topology has no cycles like a tree and it is called Destination Oriented Directed Acyclic Graph (DODAG). The choice of the preferred parent is based on the rank computed from the objective function which has as parameters the metric published by the DIO messages. These messages are sent by each node to its neighbors. The rank of each parent must be lower than the rank of the child node to prevent loops. If a node does not receive any DIO messages or wants to attach to a new DODAG for the first time, it sends a message of solicitation (DIS) to its neighbors which can answer with a DIO message if they want.

When a node wants to send a packet to a node other than the sink, there is a mechanism which consists in the transmission of DAO messages to construct the downward routes. Each node sends DAO messages to its preferred parent. If a node adds its prefix (i.e. address) to the DAO message, this is called the non storing mode and the sink must collect all the prefix of the nodes. However, if each node memories the
prefix of its sub-DODAG, that means it has enough memory and this is called storing mode. Whenever a node chooses to leave its DODAG or to re-attach to another preferred parent, it poisons its sub-DODAG by sending a rank of INFINITE-RANK. This is called local repair. There is another mechanism that regulates the inconsistency called global repair in which the sink node sends new DODAG sequence number in the DIO messages and then each node refreshes its rank.

RPL regulates the transmission of DIO messages and reduces the control traffic overhead by the trickle timer algorithm. Before starting the timer, the interval \( I_{\text{min}} \) is set to a value in the range \( \left[ \frac{I}{2}, I \right] \). The counter variable \( c \) helps to keep track of the system condition whether it is inconsistent or consistent. If it is consistent, the counter is incremented. Otherwise, it is reset. The value of \( c \) must always be less than \( k \) which is called the redundant constant. The trickle is reset to \( I_{\text{min}} \) when the timer expires. Otherwise, it is doubled.

In mobile environment, there are many disconnections due to the mobility of sensor nodes. This inconsistency leads to frequently re-initialize the interval of DIO transmission. That is why the trickle timer algorithm is not adapted to such mobile network topology.

### IV. PROPOSED MOBILITY MANAGEMENT

There are two studies in the literature closely related to our work in the sense that they have adapted the trickle timer algorithm to mobility. These protocols are MERPL [7] and RPL with reverse trickle [8]. Both of them use assumptions that are not always true. In fact in MERPL, it is supposed that the movement of the mobile node in the future depends on its mobility in the past. In RPL with reverse trickle timer, it is assumed that when a mobile connects to a new parent, it has a high chance to remain attached to this parent for a long time. Then, the more the mobile node remains connected to the same parent, the more it is likely to move outside the coverage of the parent. However, the mobile node may be at first in motion and suddenly stops its movement.

By considering the random mobility and trajectory of the sensor node, a new timer algorithm is designed in this paper. First, the minimum \( \text{Time to Leave (TL)} \) that needs a mobile node to leave the radio range of its preferred parent is computed. Then, based on the old \( TL \) and the new computed \( TL \), the number of control messages transmitted is dynamically adjusted.

In addition to that, the RPL protocol is improved by modifying the selection of the preferred parent. A node is not selected only based on its Expected Transmission Count (ETX) but also based on its stability. A node chooses its preferred parent based on a low ETX and a high Received Signal Strength Indicator (RSSI).

#### A. Preferred parent selection

The node’s relative movement is detected by analysing the difference between consecutive RSSI values [11]. As varying channel conditions and noise usually induces some variations in the RSSI values, a variation threshold is defined. At first, every node attaches to the DODAG according to its objective function. Then after receiving 5 DIO messages from the same node, it determines the moving average RSSI by computing the unweighted mean of the previous received packets. This aims to smooth RSSI values that can be affected by environment fluctuations and communication interferences [12].

Every node chooses a single preferred parent with the lowest rank and the highest mean RSSI. A minimum RSSI threshold of -80 dbm is also set. DIO messages with a RSSI value lower than the chosen threshold are automatically discarded. Two parameters \( a \) and \( b \) are introduced in the Rank calculation. The purpose of these parameters is to assure a trade-off between choosing as a preferred parent a node with a minimum ETX and a node with a high RSSI. The procedure to select the preferred parent is described in the Algorithm 1.

#### Algorithm 1 Preferred parent selection

```
repeat
    Send DIS
until Receive DIO with ETX and RSSI
Compute Rank = \( a \cdot \text{ETX} - b \cdot \text{RSSI} \)
select a preferred parent with a lower rank
```

#### B. The adaptive timer

To support sensor mobility, an adaptive timer for mobile sensors which depends on the distance between the mobile node and its preferred parent is proposed. The different positions of each mobile node and its preferred parent are sent with DAO messages. A new field of position is added to DAO.

Each mobile node computes the Euclidean distance \( E \) which represents the distance between mobile node and its preferred parent as shown in the Figure 1. The minimum remaining distance \( d \) that the mobile node needs to leave the radio range of its preferred parent is also computed by making the difference between the radio range of its preferred parent \( R \) and the Euclidian distance \( E \) (See Figure 1). Theses two variables can indicate the direction followed by each mobile node and the minimum time needed to move out the range of its preferred parent.

Each mobile node determines the minimum Time to Leave \( TL \) which represents the minimum time to move out the radio range of its preferred parent. This time depends on the velocity and the direction of mobile node. To compute the \( TL \), the remaining distance \( d \) should be divided by the speed of nodes. However in this work, each node is assumed to follow the random way mobility model which means that each node has a maximum velocity \( V_{\text{max}} \), a minimum velocity \( V_{\text{min}} \) and a pause time. To get the minimum \( TL \), the remaining distance \( d \) should be divided by the maximum velocity as shown in the Equation (1).

\[
TL = \frac{R - E}{V_{\text{max}}} = \frac{d}{V_{\text{max}}}
\] (1)

In the proposed adaptive timer algorithm, two times to Leave \( TL_1 \) and \( TL_2 \) are calculated. During the first Time to
Leave $TL_1$, the mobile node $M_{node}$ sends periodically DIO message using the maximum interval $I_{max}$. If an inconsistency is detected, $M_{node}$ resets its trickle timer. When $TL_1$ expires, $M_{node}$ computes the second remaining Time to Leave $TL_2$. If $TL_2$ is lower than $TL_1$ and $TL_1$ is lower than threshold, $M_{node}$ is suspected to disconnect from the radio range of its preferred parent. Hence, $M_{node}$ sends immediately DIS message with a flag set to 1 to its preferred parent and connects to a new parent. This DIS message initializes the DTSN which triggers transmitted DAO messages from the sub-DODAG of the preferred parent. Otherwise, if $TL_2$ is greater than $TL_1$, $M_{node}$ sets its DIO interval to $I_{min}$. If $TL_2$ is greater than $TL_1$, $M_{node}$ sets its DIO interval to the maximum value $I_{max}$. This procedure is repeated until $M_{node}$ disconnects from its preferred parent.

The adaptive timer procedure is illustrated by the Algorithm 2.

Algorithm 2 The adaptive timer algorithm

```plaintext
if DIS reception and $MF = 1$ then
    initialize $I_{min}$, $I_{max}$ and threshold
    compute $TL_1$
while $M_{node}$ is attached to its preferred parent do
    while $Clocktime + I_{max} < TL_1$ do
        $I = I_{max}$
        Schedule next DIO in $I$
    end while
    Compute $TL_2$
    if $TL_2 < TL_1$ and $TL_1 < Threshold$ then
        send DIS and connect to a new preferred parent
    end if
    if $TL_2 < TL_1$ and $TL_2 > Threshold$ then
        while $Clocktime + I_{min} > TL_2$ do
            $I = I_{min}$
            Schedule next DIO in $I$
        end while
    end if
    if $TL_2 > TL_1$ then
        while $Clocktime + I_{max} < TL_2$ do
            $I = I_{max}$
            Schedule next DIO in $I$
        end while
    end if
end while
TL_1 = TL_2
end if
```

C. Overview of the modified RPL protocol (mod-RPL)

At the beginning, each node sends in multicast DIS messages in order to discover new DODAGs in its neighbourhood. Once it receives a DIO, it chooses the preferred parent with the lowest rank based on the ETX and RSSI and then joins the DODAG. The mobile node regulates the transmission of DIO messages based on the distance to its preferred parent. Then, it computes the adaptive timer described in the Algorithm 2.

If the preferred parent of the mobile node can move, the $TL$ would be divided by 2. This can be explained by the fact that our purpose is to determine the minimum time that the mobile node takes to leave the radio range of its preferred parent. If the preferred parent and mobile node move in the opposite direction with a maximum velocity, the euclidian distance between the two nodes $E_1$ increases and the remaining distance $d_1$ decreases. In this case, it is assumed that $d_1 = \frac{d_2}{2}$ and hence $TL_2 = \frac{TL_1}{2}$.

If the mobile node remains in the radio of its preferred parent, it is not necessary to send frequently DIO messages to its neighbors. However, if it has a high probability to disconnect from its preferred parent, it may poison its sub DODAG which will become floating. Consequently, the transmission of DIO message will become frequent. In one hand, this solution aims to reduce the overhead. In other hand, it seeks to prevent from packet loss if the mobile node has a high probability to disconnect from its preferred parent. In fact, the mobile node sends a specific DIS message with a flag $MF$ set to 1 to inform its preferred parent about its disconnection.

V. Simulation results

Simulations were conducted on Cooja simulator which uses the ContikiRPL [13]. Cooja is the default simulator of Contiki operating system [14] which implements RPL protocol and can emulate the behaviour of real motes. Contiki was compiled for the Tmote Sky platform [15] which is based on MSP430 micro-controller with 10 KB of RAM and IEEE 802.15.4 chipcon wireless transceiver.

In order to evaluate the performance of mod-RPL, extensive simulations were performed and compared with native
RPL [4], MERPL [7] and RPL with reverse trickle [8]. Random network topologies were used in the simulations where ten fixed sensor nodes with an increasing number of mobile nodes were deployed. The velocity of mobile nodes was varied from 0 to 10 m/s.

Table V depicts the different parameter values used in the simulations. The two parameters \( a \) and \( b \) used in the rank calculation were respectively set to 2 and 1. This aims to achieve a trade-off between choosing as a preferred parent a node that has a minimum ETX and a high RSSI. A factor of 2 was set to ETX and a factor of 1 was set to RSSI in order to have as a preferred parent a node not far from the sink and at the same time is not moving fast (\( i.e. \) more stable).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network simulator</td>
<td>Cooja under Contiki</td>
</tr>
<tr>
<td>Simulation time</td>
<td>1 hour</td>
</tr>
<tr>
<td>Type of emulated nodes</td>
<td>Tmote Sky</td>
</tr>
<tr>
<td>Chipcon radio, MAC, PHY</td>
<td>IEEE 802.15.4</td>
</tr>
<tr>
<td>Frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Data packet period</td>
<td>60 seconds</td>
</tr>
<tr>
<td>Packet size</td>
<td>128 Bytes</td>
</tr>
<tr>
<td>Number of sink</td>
<td>1</td>
</tr>
<tr>
<td>Transmission range of nodes</td>
<td>50 m</td>
</tr>
<tr>
<td>( I_{\text{min}} )</td>
<td>12 Clocktime</td>
</tr>
<tr>
<td>( I_{\text{max}} )</td>
<td>20 Clocktime</td>
</tr>
<tr>
<td>Threshold</td>
<td>3 Clocktime</td>
</tr>
<tr>
<td>Rank</td>
<td>( a = 2, b = 1 )</td>
</tr>
</tbody>
</table>

The performance of the different protocols were analyzed in term of packet delivery ratio, overhead, power consumption and end-to-end delay.

A. Overhead

Overhead is the number of control packets transmitted to maintain links in the DODAG. The overhead includes DIS, DIO, DAO, DAO-ACK packets. The rate of control messages should be limited to reduce energy consumption of sensor nodes.

Figure 2 shows the impact of varying the number of mobile nodes on the amount of control message exchanged in the network. The overhead increases when the number of mobile nodes grows because nodes need to inform their neighbourhood about the changes occurred in the DODAG by transmitting more control messages. The results show that mod-RPL offers the lowest overhead compared to other protocols. Because with mod-RPL, mobile nodes send rarely DIO messages when they are inside the radio range of their parents. However with reverse trickle timer, the preferred parent sends at first frequently DIO messages. In addition to that, mobile nodes move randomly which resets each time the trickle timer interval. Hence the nodes send frequently DIO messages. With MERPL, mobile node movement does not necessarily depends on its past. So, mobile node disconnections often occur which leads to frequently resetting the trickle timer.

B. Average Delay

Figure 3 shows that the average delay increases with the number of mobile nodes. In general, RPL with trickle timer has the lowest delay followed by mod-RPL. In fact with mod-RPL, a node chooses its preferred parent based on the average RSSI and ETX. In case of a DODAG with an increasing number of fixed nodes, a node choose its preferred parent with the lowest ETX, which decreases the delay in the network. However, if the number of mobile nodes increases a node will have a high probability to choose as a preferred parent a node that has not the best ETX but the highest RSSI. This choice enhances stability in network but increases the average delay.

C. Average Power Consumption

Figure 5 represents the power consumption with an increasing number of mobile nodes. In general, the power consumption grows with the number of mobile nodes because nodes need to exchange more control packets to maintain the
DODAG. RPL with reverse trickle consumes less power than mod-RPL and MERPL because it does not have a computation cost. In fact, mod-RPL and MERPL make respectively computations to determine the Time to Leave and different DIS intervals which leads to an extra power consumption. However, the power expended with mod-RPL can be reduced by optimizing the calculation conducted in the used algorithm.

![Fig. 4. Total power consumption in random network topology with different number of mobile nodes.](image)

**D. Average Packet Delivery Ratio**

Figure 5 shows that mod-RPL has the highest Packet Delivery Ratio (PDR) compared to native RPL, MERPL and RPL with reverse trickle. This can be explained by the fact that in mod-RPL neighbors including preferred parents that are directly connected to a mobile node are aware of its disconnection. Thanks to the proposed adaptive timer mechanisms that future disconnections can be early detected. Consequently, neighbors send packets only before the end of the leaving time which prevents from packet loss. Although in RPL with reverse trickle mobile nodes can quickly detect the unreachability of their current parents, there are some cases where stability of mobile nodes increases and hence the reverse trickle is mismatching. With MERPL, mobile nodes’ movement does not necessarily depend on its past. Therefore with this assumption, mobile nodes can not always detect the unreachability of their current parents.

![Fig. 5. Average packet delivery ratio in random network topology with different number of mobile nodes.](image)

**VI. Conclusion**

In this paper, a modification of RPL protocol is presented to fit with random node mobility requirements. The new protocol introduces a new adaptive timer algorithm for mobile nodes that regulates the transmission of DIO and DIS messages. The proposed algorithm exploits the leaving time of mobile node in order to prevent from disconnections and packet loss. The simulation results show that our enhanced protocol offers better performance than native RPL, MERPL and RPL with reverse trickle in term of overhead and packet delivery ratio. As future work, it will be interesting to further enhance the performance of mod-RPL and regulate node’s idle listening in order to save more energy.

**REFERENCES**