Article:

Kharrufa, H, Al-Kashoash, H and Kemp, AH (2018) A Game Theoretic Optimization of RPL for Mobile Internet of Things Applications. IEEE Sensors Journal, 18 (6). pp. 2520-2530. ISSN 1530-437X

https://doi.org/10.1109/JSEN.2018.2794762

A Game Theoretic Optimization of RPL for Mobile Internet of Things Applications

Harith Kharrufa, *Student Member, IEEE,* Hayder Al-Kashoash, and A.H. Kemp, *Senior Member, IEEE*

***Abstract*—The presence of mobile nodes in any wireless network can affect the performance of the network, leading to higher packet loss and increased energy consumption. However, many recent applications require the support of mobility and an efficient approach to handle mobile nodes is essential. In this paper, a game scenario is formulated where nodes compete for network resources in a selfish manner, to send their data packets to the sink node. Each node counts as a player in the non-cooperative game. The optimal solution for the game is found using the unique Nash equilibrium (NE) where a node cannot improve its pay-off function while other players use their current strategy. The proposed solution aims to present a strategy to control different parameters of mobile nodes (or static nodes in a mobile environment) including transmission rate, timers and operation mode in order to optimize the performance of RPL under mobility in terms of packet delivery ratio (PDR), throughput, energy consumption and end-to-end-delay. The proposed solution monitors the mobility of nodes based on received signal strength indication (RSSI) readings, it also takes into account the priorities of different nodes and the current level of noise in order to select the preferred transmission rate. An optimized protocol called game-theory based mobile RPL (GTM-RPL) is implemented and tested in multiple scenarios with different network requirements for Internet of Things applications. Simulation results show that in the presence of mobility, GTM-RPL provides a flexible and adaptable solution that improves throughput whilst maintaining lower energy consumption showing more than 10% improvement compared to related work. For applications with high throughput requirements, GTM-RPL shows a significant advantage with more than 16% improvement in throughput and 20% improvement in energy consumption.**

***Index Terms*—Routing, WSN, IoT, RPL, mobility, game theory, 6LoWPAN, IoMT.**

I. Introduction

The Internet of Things (IoT) evolution is leading to numerous opportunities in different applications including robotics, healthcare, smart environments, sports monitoring, animal tracking, smart agriculture and many other fields on both small and large scales. The Internet engineering task force (IETF) has developed a number of standards in the recent years to enable small resource-limited sensor nodes to communicate effectively. The IETF routing over low-power and lossy networks (ROLL) working group provided protocol specifications for the routing protocol for low-power and lossy networks (RPL). Using RPL, IPv6 nodes can communicate at the network level and be easily connected to the Internet, RPL is thus considered the routing protocol of IoT.

RPL uses IPv6 over low-power wireless personal area networks (6LoWPAN) as an adaptation layer to allow the full integration of wireless sensor networks (WSNs) and the Internet. 6LoWPAN resides between the IEEE 802.15.4 MAC and the Internet protocol (IP) in the network layers of the IoT protocol stack, it is responsible for packet fragmentation and reassembly. This allows transmission of IP packets with a maximum transmission unit (MTU) of 1280 bytes over IEEE 802.15.4 links which allow a maximum of only 127 bytes for data.

RPL was originally designed for static networks, once the connections are established, it assumes that the network is in a steady state and does not take mobile nodes into account. There are many efforts to enhance RPL and many are successful in creating new versions of RPL that take into account the presence of mobile nodes. However, none of these efforts consider analysing and optimizing the efficiency of RPL in a mobile environment with regard to throughput, energy consumption and end-to-end delay. Therefore, in this paper, an analytical model is provided with a proposal for a game theoretic design of RPL (GTM-RPL) using a variable transmission rate to achieve higher packet delivery ratio (PDR), lower end-to-end delay and better throughput whilst maintaining efficient energy consumption. To achieve this, a game is designed for nodes competing to send data in a mobile environment, where mobility itself serves as an involuntary action that influences decision making in all affected nodes. The payoff function is defined to assess the profit gained from increasing data transmission rate (the utility function) against the cost induced by the presence of mobile nodes (mobility function). Other factors are also taken into account in formulating the payoff function including the priority of nodes (priority function) and the energy consumption (energy function) . In order to prove the presence of at least one Nash equilibrium, a discussion and analysis are provided along with the optimal solution of the game. Then, a proposal of a novel GTM-RPL based on this design and a performance evaluation in different IoT application scenarios are provided and tested using Cooja over Contiki OS in a simulation environment.

The rest of the paper is organised as follows: Section II provides an overview of research directed towards enhancing RPL in the presence of mobility. Section III provides a description of the native RPL and the proposed GTM-RPL with a discussion on the related aspects of the protocol and the formulation of the optimization game. Section IV presents

Harith Kharrufa and A. H. Kemp are with the Electronic and Electrical Engineering School, University of Leeds, Leeds LS2 9JT, U.K. (e-mail: elhdy@leeds.ac.uk; a.h.kemp@leeds.ac.uk).

H. A. A. Al-Kashoash is with Technical Institute/Qurna, Southern Technical University, Basra, Iraq (e-mail: hayderaam@gmail.com).

the simulation settings and results, and provides a discussion to compare GTM-RPL with relevant protocols in different scenarios. Finally, Section V discusses the conclusions from the paper.

II. Related Work

RPL is a standardized routing protocol for low power and lossy networks (LLNs) and is considered the standard routing protocol for IoT applications [1]. There are many efforts to improve and create enhanced versions of RPL taking advantage of its flexible and scalable design. Since one of the obvious disadvantages of using RPL is that it lacks mobility support, several researchers focus on providing solutions to accommodate mobile nodes.

The authors in [2] included a mobility flag in the DIO message to distinguish static nodes, mobile nodes are then given a higher cost in the parent selection process. This protocol offers a higher PDR and better network stability, however it does not include any metrics in the parent selection process and it changes the RPL standard by adding the mobility flag to DIO messages making it incompatible with the native RPL.

In [3], the authors added geographical information to the routing metrics to find the direction of the node and to choose the parent node in the forward direction. This approach minimizes the number of dissociations which in turn improves the stability of the network, but as it was designed for vehicular networks, it assumes that nodes do not change their direction. In addition to that, the results were based on a single mobile node collecting data from static road signs.

In [4], the authors provided analysis of RPL under mobility using a new algorithm for the trickle timer. The protocol uses a reverse trickle timer for mobile nodes, this timer decreases the interval until it reaches its minimum value. This protocol improves recovery of disconnected mobile nodes but it requires the user to preconfigure nodes with a mobility flag since it has no way of detecting mobility on its own.

The authors in [5] introduced a corona structure with RPL (Co-RPL). This mechanism divides the DODAG into circular ”coronas” around the root node to allow dissociated nodes to seamlessly re-join the DODAG. It also uses a novel objective function based on fuzzy logic FL-OF, this function uses link quality, hop count, delay and residual energy as routing metrics. This protocol improves PDR, delay and energy consumption compared to the native RPL but it only supports one mobile node moving at low speed and cannot accommodate multiple mobile nodes in a dynamic network.

An interesting approach is introduced by [6] for healthcare applications, it assumes a network with both mobile and static nodes; a sound assumption for many healthcare applications. Mobile nodes are configured to only act as leaf nodes meaning that they do not send DIO messages and cannot serve as routers. This approach allows mobile nodes to join the DODAG but not to advertise themselves leading to a better stability for the network. In this approach, the authors only evaluate the performance of RPL with these settings but do not actually change anything in its design.

One of the most popular protocols for mobile RPL is mRPL [7] which is designed for mobile IoT applications. This protocol significantly improves the operation of RPL in the presence of mobile nodes by introducing four additional timers to detect and manage mobility. The connectivity timer is responsible for detecting a loss of connectivity to the parent node. The mobility detection timer uses the average received signal strength indication (ARSSI) to assess the reliability of the connection. The hand-off timer is responsible for allocating an adaptive short period that is sufficient for sending bursts of DIS and receiving DIO replies in order to reduce the hand-off delay. The reply timer is responsible for sending replies to the mobile nodes using an adaptive period to minimize collisions. mRPL considers ARSSI the only metric for routing and does not consider using other metrics resulting in many unnecessary hand overs. One of the improvements to mRPL was introduced in [8] as mRPL++. It suggests using other routing metrics in the objective function as the product of ARSSI and the ratio between routing metrics of the potential parent node and the current one. This protocol does not add much to mRPL and is sometimes even less efficient in terms of PDR, delay and energy consumption.

The authors in [9] proposed D-RPL for multihop routing in dynamic IoT applications, aiming to improve the operation of RPL in mobile environments with dynamic requirements. D-RPL uses some of the features of mRPL in addition to an adaptive timer that works as a reverse-trickle timer when mobility is detected. It also includes routing metrics in the decision making to minimize the number of unnecessary hand overs while maintaining high responsiveness and smooth transitions.

The novel contributions of this paper are: (i) Improving and optimizing mobility management in RPL using a game theoretic approach. (ii) Introducing an adaptive transmission rate that depends on the conditions of the network and the availability of resources. (iii) Using a RSSI and link quality indicator (LQI) to assess the level of noise and the mobility conditions at each node. (iv) Adding cost functions to reflect on energy efficiency and priority, leading to an optimum transmission rate that matches the network conditions and application requirements.

III. Game-Theory Based Mobile RPL (GTM-RPL)

A. RPL description

RPL is an IPv6 routing protocol that was designed by the IETF ROLL working group for low-power and lossy networks (LLNs), it operates on the IEEE 802.15.4 standard using 6LoWPAN as an adaptation layer. This protocol creates a multi-hop hierarchical topology for nodes, where each node can send data to its parent node which in turn forwards it upward until it reaches the sink or gateway node. It successfully and efficiently manages data routing for nodes that have restricted resources. It minimises the number of transmitted control messages by introducing the trickle timer [10] with an exponentially incremented interval to manage transmission of control packets. The idea of the trickle timer is built on the assumption that the network does not need control messages except in the initial phase of establishing connectivity. Although this is true in a static network, mobile nodes make it impractical to use the trickle timer in its original specifications. The main parameters of the trickle timer are I_{min}, $I_{doubling}$ and I_{max}.

$$I_{min} = 2^n \quad (1)$$

$$I_{max} = 2^{n+I_{doubling}} \quad (2)$$

The interval n produces I_{min} (ms) which is the initial and minimum interval size of the trickle timer as shown in equation (1). $I_{doubling}$ decides I_{max} (ms) which is the maximum interval size of of the trickle timer as shown in equation (2). The selection of these values is critical even in a static network because they directly affect PDR, end-to-end delay and energy consumption of the network. High intervals lead to low responsiveness to network's inconsistencies including those caused by node mobility, while low intervals mean higher overhead leading to shorter lifetime for the network.

RPL builds the topology of the network based on a Directed Acyclic Graph (DAG) with no outgoing edges so that no cycles can exist. Every DAG is routed towards one or more DAG roots forming a Destination-Oriented DAG (DODAG). The DODAG is built using the predefined objective function which contains the metrics for route selection. RPL maintains connectivity using a number of control messages. The DODAG Information Object (DIO) carries information including the DODAG-ID and the rank to allow other nodes to discover the DODAG. The Destination Advertisement Object (DAO) contains the RPL instance ID that was learnt from the DIO and it is sent from the child node to the parent node or the DODAG root. The DODAG Information Solicitation (DIS) is used to request a DIO from an RPL node.

There are currently two objective functions presented by the IETF, the first one is Objective Function zero (OF0) [11] which is a simple and basic objective function where nodes select a parent node based on its rank in the DODAG. It does not consider any other routing metrics and it is designed as a general objective function that allows implementations of other objective functions. The second one is theMinimum Rank with Hysteresis Objective Function (MRHOF) [12] which is based on routing metric containers. It transmits the metric container in the DIO message and it uses the Expected Transmission Count (ETX) to calculate the rank of the node, it also supports residual energy as a routing metric.

A node using RPL starts its operation by waiting for a DIO message, the probability that a node receives this message in a given time depends on the number of neighbouring nodes and their trickle settings. Once the node receives a DIO, it sends a DAO message to the DODAG root and moves to the active state. Depending on the application, the node transmits or relays data towards the sink node and expects to receive periodic DIO messages from its parent node. These messages are sent periodically depending on the state of the network and the settings of the trickle timer as shown in equations (1) and (2).

The transition states of RPL are shown in Fig 1. The main goal is to optimize RPL so that a node can have a high probability of (b, c and d) and a low probability of (a). When an RPL node starts, it waits for a DIO and the probability that

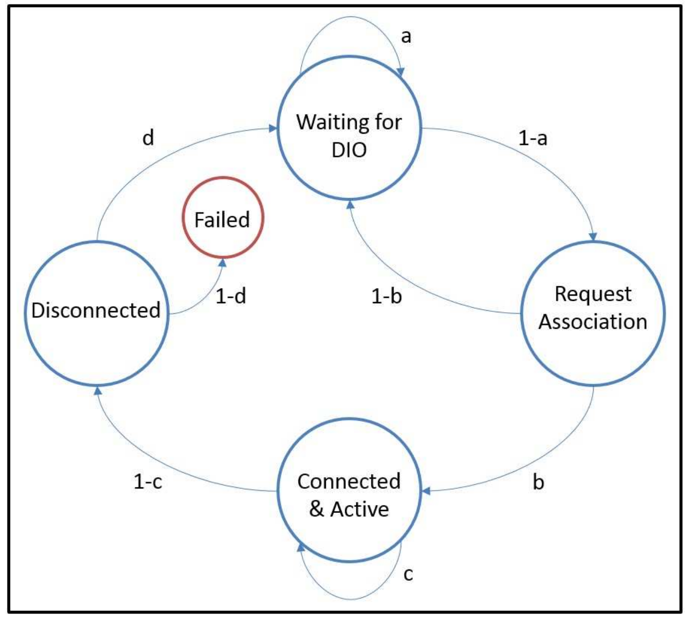

Fig. 1. Markov chain for RPL nodes

it stays in that state is represented by (a). If this node receives a DIO, then it requests association from the potential parent node and this is given a probability of (1-a). The probability of a successful association is represented by (b) and therefore, the probability of a failed request is (1-b). Once the node is successfully connected, it starts sending data towards the sink and this is denoted by a probability (c). In this state, there is a (1-c) probability of dissociation due to any reason including node mobility. Finally, there is a (d) probability that the node is still in operation and in this case, it can restart the cycle and wait for another DIO. In turn, if the energy is depleted, the node fails and cannot resume operation until it is fitted with new batteries and that is represented by a probability (1-d). With the presence of mobile nodes in the network, adaptive settings need to be added to RPL and for that reason, a non-cooperative game is formulated where nodes compete for network resources taking into account the requirements of the application and the conditions of the network.

Although the application scenarios give an indication of a cooperative behaviour, nodes are competing to send data at higher transmission rates, causing higher levels of noise. A node that increases its transmission rate, is maximising its utility function but is also negatively affecting the utility function of other nodes. This means that increasing transmission rate will increase the payoff of the node itself, but not necessarily the collective payoff of all players. For these reasons, the game is considered a non-cooperative game with a goal to maximise gain and minimise cost for the whole network.

B. GTM-RPL Game Formulation

Assuming a network with one static sink node that serves as a gateway, a number of static nodes to ensure better coverage and a number of mobile sensor nodes as shown in Fig 2.

Players $P = p_1, p_2, \ldots, p_n$ are competing to send data packets to the sink node while playing the mobility man-

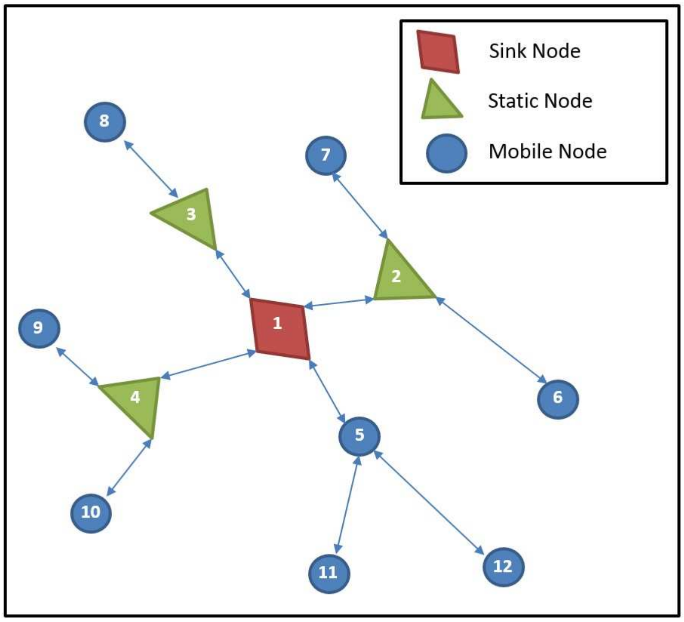

Fig. 2. RPL topology

agement game. In game theory, each action performed by a player affects the utility function of other players, actions include changing data rate, parent node, trickle settings and transmission power. The following rules define the game: (i) Each node p_k can send data at a rate of [0 , λ_{max}]. (ii) Mobile nodes have user-defined priorities $R = r_1,\ldots,r_k,\ldots,r_n$ where r_k is the priority of node p_k $\forall k \in N$, nodes with higher priority assume lower cost for energy consumption to allow them to send data at higher rates. (iii) All nodes share an application specific mobility metric Mm that reflects the expected mobility intensity in a specific application, and a density metric Dm that depends on the number of nodes, coverage area of each node and total simulation area, if these two metrics are not defined by the user, they are assumed Mm^o and Dm^o respectively. (iv) Each node can measure the RSSI of each message at the MAC layer to compute the link quality (LQ) at a given time (t). (v) Sensor nodes have limited resources with the exception of the sink node. (vi) All nodes use Contiki OS with 6LoWPAN adaptation layer and inherit their benefits and restrictions. The mobility management game is defined by $\Gamma = (N, (S_k)k \in N, (\phi_k)k \in N)$, where N is the number of players, S_k is a vector of the possible strategies for player P_k, and ϕ_k is the payoff function for player P_k. The payoff of each player represents the cost that a node P_k must endure for taking an action A_k.

1) Players: represent the sensor nodes in the same collision space of the network, $(P_1, \ldots, P_k, \ldots, P_n), \forall k \in N$.

2) Strategies: each node has a set of possible actions $A = (A_1, \ldots, A_k, \ldots, A_n) \forall k \in N$. Where $A_k = [0, \lambda_{max}]$ represents the strategy space for player P_k and thus $A = \prod_{k=1}^{N} A_k$.

3) Payoff function: $\phi_k(A_k)$: defines the total cost for node P_k to send data at a rate of λ_k to the sink node in a mobile environment. The payoff function is defined to include the profit (the utility function), the cost induced by mobility, the energy cost and the node priority cost as follows:

- Utility Function $U_k(A_k)$: represents the profit of player P_k for using the strategy A_k. This function reflects the gain of increasing transmission rate λ_k as each node tries to maximise its throughput. In order to make sure that the utility function is concave and its second derivative is always negative, the utility function is defined as:

$$U_k(a) = \alpha \log(\lambda_k + C) \tag{3}$$

Where α is a user defined factor and C is a safety constant to make sure that there is always a defined value for the utility function, otherwise at $\lambda = 0$, the value goes to infinity. For each player, the goal is to increase transmission rate to maximize the utility function and thus the profit, taking into account the negative effects that may come with that, this trade-off is explained in the other cost functions.

- Mobility Function $M_k(a_k, a_{-k})$: this function gives a measure of the cost incurred by the presence of mobility, where a_{-k} is the actions available for all players except $P_k(P_1, \ldots, P_{k-1}, P_{k+1}, \ldots, P_n); k \in N$. In order to have a measure of mobility, $(ARSSI)$ and LQI are used to evaluate the link quality cost (LQ) as in [9]. Also, an estimated mobility metric that is application specific is used to indicate the mobility level for a given application. The calculation of this metric depends on the mobility scenario. In the simulations, the random waypoint mobility model is used because it fairly reflects the actual mobility behaviour in WSNs and IoT applications [13][14].

$$M_k(a_k, a_{-k}) = \beta \ Mm \ LQ \ \lambda \tag{4}$$

Where β is a factor that can be changed in accordance with the preference of the user and the type of the application. Mm is the mobility metric and it is estimated according to the mobility scenario. In order to calculate Mm the following formula is used [15]:

$$Mm = \frac{1}{|N|} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{T} \int_{0}^{T} |V_i(t) - V_j(t)| \ dt \tag{5}$$

Where N is the number of node pairs in the network and is equal to the total number of nodes in the RPL topology. T is the total runtime in seconds. $V_i(t) - V_j(t)$ is the difference in speed between nodes i and j at time t. This metric is not calculated based on the actual movement of nodes because it is not possible to predict, but rather based on a generated mobility scenario using Bonnmotion [16] , a free and widely used tool for mobility scenario generation. In order to calculate LQ, extensive simulations are conducted to measure the effect of different LQ levels and the points where they can be assumed reliable in terms of packet loss and transmission delay.

- Energy function $E_k(a_k, a_{-k})$: energy consumption is one of the most important factors in many IoT applications, especially in cases where the cost of replacing batteries is high. In any application, lower energy consumption means better life span for the node itself and for the whole network. $ARSSI$ and DIS messages are used to control the trickle timer as in [9][7] and minimize the energy consumed due to control messages. However, with regards to optimizing throughput, limitations arise from the increased energy consumption caused by sending data packets to the sink node. Higher data rate means more packet transmissions and thus higher energy consumption. Another important factor is the density of the network, higher density means more data is relayed which incurs additional packet transmissions for all nodes. The density of the network also causes higher congestion at the relay nodes leading to higher energy consumption for relaying data and retransmitting lost packets.

$$E_k(a_k, a_{-k}) = \gamma \ Dm \ \lambda \tag{6}$$

Where γ is the user defined weight given for energy saving requirements, Dm is the density metric of the network. In order to express the level of density in a network, this simple formula is used [17]:

$$Dm = \frac{|N|\pi T_r^2}{A} \tag{7}$$

Where N is the number of nodes, T_r is the transmission range for each node and A is the deployment area. In the simulations, it is assumed that the deployment area has a good coverage giving a density metric $Dm > 1$.

- Priority function $Pr_k(a_k)$: In many IoT applications, some nodes can be of higher importance than others. For example, in a healthcare application, a node that monitors the well being of a patient and informs a member of staff in case of an emergency (fall detection, health risk, etc.) is usually given a higher priority than nodes used for controlling room temperature. The priority of nodes is set by the user to the preferred level, otherwise nodes assume $Pr_k = Pr_k^0$ as the default priority.

$$Pr_k(a_k) = \delta \ pr_k \ \lambda \tag{8}$$

Where δ is the user defined weighing factor, pr_k is the priority of node $k, \forall k \in N$.

The factors α, β, γ and δ are added to give higher flexibility to the design of GTM-RPL, allowing the user to customize it according to the application demands and requirements. For each player $P_k \forall k \in N$, the payoff function can be declared as:

$$\phi_k(A_k) = \alpha \log(\lambda_k + C) - \beta \ Mm \ LQ \ \lambda - \gamma \ Dm \ \lambda - \delta \ pr_k \ \lambda \tag{9}$$

In order to find a solution to the game $\Gamma = (N, (S_k)k \in N, (\phi_k)k \in N)$, a proof that it has a unique Nash equilibrium is required, this means that each player can reach an optimal strategy $s_k^* = \lambda_k^*$ where it has no incentive to change its strategy given that all other players maintain their current strategies.

Theorem 3.1: The formulated game is a concave n-person game and it has at least one Nash Equilibrium.

Proof: The strategy vector for player P_k can be represented by $S_k = [0, ..., \lambda_k^{max}]$, it is clear that the strategy set of player P_k is closed and bounded meaning that the set S_k is compact $\forall k \in N$. Consider x, y to be two points in the strategy vector S_k in a Euclidean space where $S = \prod_{k=1}^{n} S_k$, the strategy set S_k is convex if for any $x, y \in S_k$ and $\eta = [0, 1]$, $\eta x + (1 - \eta) y \in S_k$.

The Hessian matrix of the payoff function $\phi_k(A_k) = \alpha \log(\lambda_k + C) - \beta \ Mm \ LQ \ \lambda - \gamma \ Dm \ \lambda - \delta \ pr_k \ \lambda$ can be defined as:

$$H = \begin{bmatrix} \frac{\partial^2 \phi}{\partial \lambda_1^2} & \frac{\partial^2 \phi}{\partial \lambda_1 \partial \lambda_2} & \cdots & \frac{\partial^2 \phi}{\partial \lambda_1 \partial \lambda_n} \\ \frac{\partial^2 \phi}{\partial \lambda_2 \partial \lambda_1} & \frac{\partial^2 \phi}{\partial \lambda_2^2} & \cdots & \frac{\partial^2 \phi}{\partial \lambda_2 \partial \lambda_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \phi}{\partial \lambda_n \partial \lambda_1} & \frac{\partial^2 \phi}{\partial \lambda_n \partial \lambda_2} & \cdots & \frac{\partial^2 \phi}{\partial \lambda_n^2} \end{bmatrix} \tag{10}$$

By applying the second derivative test on the payoff function ϕ_k, it is clear that the leading principal minor of the Hessian matrix is negative definite at λ meaning that it reaches a local maximum at λ as shown in equation (11) [18].

$$\frac{d^2}{d\lambda^2} = \phi_k''(\lambda) = -\frac{\alpha}{(\lambda_k + c)^2} \tag{11}$$

■

Theorem 3.2: The weighted non-negative sum $\sigma(\lambda_k, r)$ is diagonally strictly concave if the symmetric matrix $[G(\lambda_k, r) + G'(\lambda_k, r)]$ is negative definite $\forall \lambda_k \in S$ where r is a non-negative vector [19].

Proof: The weighted non-negative sum $\sigma(\lambda_k, r)$ can be written as a summation of $\phi_k(\lambda)$

$$\sigma(\lambda_k, r) = \sum_{k=1}^{n} r_k \phi_k(\lambda), \forall k \in N, r_k \geq 0 \tag{12}$$

For each fixed value of $r = (r_1, r_2, \ldots, r_n)$, a related mapping of $g(\lambda_k, r)$ is defined as gradients $\bigtriangledown_k \phi_k(\lambda_k)$.

$$g(\lambda_k, r) = \begin{bmatrix} r_1 \bigtriangledown_1 \phi_1(\lambda_1) \\ r_2 \bigtriangledown_2 \phi_2(\lambda_2) \\ \vdots \\ r_n \bigtriangledown_n \phi_n(\lambda_n) \end{bmatrix} \tag{13}$$

Where $g(\lambda_k, r)$ is the pseudo-gradient of $\sigma(\lambda_k, r)$ and $\bigtriangledown_k \phi_k(\lambda_k)$ is given by:

$$\bigtriangledown_k \phi_k(\lambda_k) = \frac{\alpha}{\lambda_k + C} - \beta MmLQ - \gamma Dm - \delta P_k^r, \forall k \in N \tag{14}$$

From $g(\lambda_k, r)$ in equation 13, its Jacobian matrix can be defined by $G(\lambda_k, r)$ as:

$$G(\lambda_k, r) = \begin{bmatrix} r_1 \frac{\partial^2 \phi}{\partial \lambda_1^2} & r_1 \frac{\partial^2 \phi}{\partial \lambda_1 \partial \lambda_2} & \cdots & r_1 \frac{\partial^2 \phi}{\partial \lambda_1 \partial \lambda_n} \\ r_2 \frac{\partial^2 \phi}{\partial \lambda_2 \partial \lambda_1} & r_2 \frac{\partial^2 \phi}{\partial \lambda_2^2} & \cdots & r_2 \frac{\partial^2 \phi}{\partial \lambda_2 \partial \lambda_n} \\ \vdots & \vdots & \ddots & \vdots \\ r_n \frac{\partial^2 \phi}{\partial \lambda_n \partial \lambda_1} & r_n \frac{\partial^2 \phi}{\partial \lambda_n \partial \lambda_2} & \cdots & r_n \frac{\partial^2 \phi}{\partial \lambda_n^2} \end{bmatrix} \tag{15}$$

Since the symmetric matrix $[G(\lambda_k, r) + G'(\lambda_k, r)] \forall k \in N, \lambda_k \in S$, is negative definite, then the weighted non-negative sum $\sigma(\lambda_k, r)$ is diagonally strictly concave and the game $\Gamma = (N, (\lambda_k)k \in N, (\phi_k)k \in N)$, has a unique Nash equilibrium [19]. ■

C. Game Solution

To find the optimum solution of the game, the payoff function $\phi_k(\lambda_k)$ needs to be maximised by choosing an optimal strategy according to the game design. The optimal transmission rate $\lambda_k^* \; \forall k \in N, \lambda_k^* \in S$ is restricted by $0 \leq \lambda_k \leq \lambda_k^{max}$. To find the solution of the game, the Lagrangian function is defined by:

$$\mathcal{L}_k = \Phi_k(\lambda_k) + u_k\lambda_k + v_k(\lambda_k^{max} - \lambda_k) \tag{16}$$

Where u_k and v_k are the Lagrange multipliers and the Karush-Kuhn-Tucker (KKT) [20] conditions for the maximization problem are:

$$\begin{gathered} u_k, v_k \geq 0 \\ \lambda_k \geq 0 \\ \lambda_k^{max} - \lambda_k \geq 0 \\ \nabla_{\lambda_k}\Phi_k(\lambda_k) + u_k\nabla_{\lambda_k}(\lambda_k) + v_k\nabla_{\lambda_k}(\lambda_k^{max} - \lambda_k) = 0 \\ u_k(\lambda_k), v_k(\lambda_k^{max} - \lambda_k) = 0 \end{gathered}$$

The solution to the game can now be solved for each player $P_k, \forall k \in N$, the outcome λ_k^* is the optimum transmission rate depending on the state of the network and the user-defined application parameters. The value of λ_k^* can be found using equation (17).

$$\lambda_k^* = \begin{cases} 0 & \text{Condition A} \\ \lambda_k^{max} & \text{Condition B} \\ \dfrac{\alpha}{\beta MmLQ + \gamma Dm + \delta pr_k} - c & \text{Otherwise} \end{cases} \tag{17}$$

where condition A and condition B respectively are:

$$\beta MmLQ + \gamma Dm + \delta pr_k \geq \alpha \tag{18}$$

$$\beta MmLQ + \gamma Dm + \delta pr_k \leq \frac{\alpha}{\lambda_k^{max} + C} \tag{19}$$

The optimum transmission rate λ_k^* is the Nash Equilibrium for that node, $\forall k \in N$. This value changes when a node moves ($RSSI$ is affected) and when an other node changes its transmission rate (LQI is affected).

D. Protocol Implementation

The proposed protocol is implemented using Contiki operating system 3.0 [21] and COOJA [22] network simulator. Algorithm 1 shows the basic operation of GTM-RPL, the main optimization point is the value of λ_k^*. In the simulation, the values of α, β, γ and δ are 4.7, 1, 0.05 and 0.1 respectively. The value of Mm is 0.725 for the simulation scenarios and the Dm is 9.42 giving a reliable coverage. The priority of nodes can take a value of [1,10] depending on the application requirements. λ_k^{max} is set to $2, 4, 8$ and 16 pkt/s and the safety factor $C = 0.1$, these values depend on the application requirement and were selected based on extensive simulations.

The value of LQ is calculated and updated at each node based on $RSSI$ and LQI and the values are mapped in Fig 3. Lower values for LQ indicate better quality as LQ represents the cost incurred due to the link quality. The initial transmission rate λ_k^0 is set at $(\lambda_k^{max}/2)$ pkt/s and then updated periodically throughout the simulation according to equation (17).

Algorithm 1 GTM-RPL operation

1: ***Initialization***:
    Set α, β, γ & δ_k
    Set λ_{max}
    Set application metrics Mm & Dm
    Set pr_k
    Initialize trickle timer $I_{min}, I_{max}, I_{doubling}$
    Set λ_0
2: ***Active mode***:
    Read $ARSSI$
    $\lambda_k^* \leftarrow equation(17)$
    If ($ARSSI_t$ + K_{RSSI} <$ARSSI_{t-1}$) then
        Send DIS to all neighbours
        $I_t^{Trickle}$ = $(I_{t-1}^{Trickle}/2)$
            If $I^{Trickle}$ <I_{min} then
            $I^{Trickle}$ = I_{min};
    Else
        Resume normal Trickle
    End

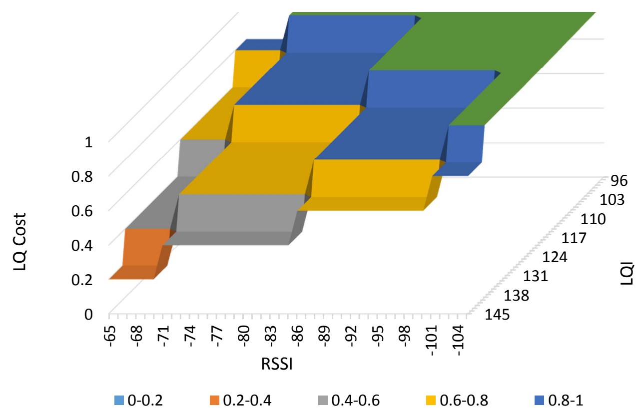

Fig. 3. Link quality

The mobility detection part of the protocol is also shown in Algorithm 1 and it uses the change in values of $RSSI$ as a mobility detection parameter. It sends multicast DIS messages to all neighbours and triggers the reverse-trickle timer to improve responsiveness and maintain connectivity.

IV. Simulation Analysis

The simulations are focussed on two healthcare applications, the first one is patient monitoring in an elderly care unit, and the second application is hospital environment monitoring. Both applications share some of the simulation parameters provided in Table I.

The proposed protocol is evaluated and compared with related protocols in terms of PDR, throughput and energy consumption using Contiki OS and COOJA simulator. The simulation uses a Tmote Sky platform which is emulated by COOJA, and a unit disk graph medium (UDGM) as the wireless channel taking into account noise levels and interference.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
λ_{max}	2, 4, 8, 16 packets / s
Packet size	64 bytes
Simulation Area	1600 m^2
Number of Nodes	11 nodes + 1 sink node
Transmission Range	20m
Mobility Scenario	Random Waypoint, 0 to 2 m/s
I_{min} / $I_{doubling}$	8 / 6
Simulation Time	1 hour
Radio	CC2420

A. Elderly Monitoring

In this application, wearable sensor nodes are attached to patients in the elderly care unit shown in Fig 4 to monitor their well being as well as information about the environment around them. These sensors read the blood pressure of patients and inform the medical staff of any abnormality. They also monitor the mobility habits of patients and provide personalized health advice. In addition to fall detection sensors that alarm the staff of any accidents. In the simulation, one sink node is used with three fixed sensor nodes to provide better coverage and eight mobile nodes attached to patients. In the simulation, the sensor nodes are all given the same priority of 5 and they compete to send periodic messages to the sink node. The results show a performance evaluation of the proposed GTM-RPL and compare it against the native RPL and mRPL. RPL has no way of managing mobility but nonetheless it is shown as a baseline for comparison. mRPL on the other hand has an excellent mobility management approach but it uses a fixed transmission rate and does not adapt to the mobility of nodes. For the sake of comparison, different transmission rates are used, 2 pkt/s and 4 pkt/s to show the performance at different settings.

Fig 5 shows the PDR as a percentage for each node, all protocols achieve high PDR (above 88%) for the first three static nodes but for mobile nodes, the native RPL goes down to around 44% at 4 ptk/s and 47% at 2 pkt/s. mRPL at 4 pkt/s achieves around 78% PDR while at 2 pkt/s reaches up to 88%. GTM-RPL achieves a similar PDR of around 88% at both transmission rates and it outperforms mRPL by more than 10% in the 4 pkt/s scenario.

Although GTM-RPL does not show an advantage against mRPL at 2 pkt/s, it is clear that mRPL unlike GTM-RPL, is not trying to optimize the transmission rate. The throughput shown in Fig 6 shows that GTM-RPL provides almost twice the size of successfully transmitted data. mRPL at (4pkt/s) is always sending at the maximum transmission rate and yet it does not show an advantage compared to GTM-RPL in terms of throughput. This is because it has a lower PDR and thus a higher number of packets are dropped before reaching the sink node.

Fig 7 shows the energy consumption (mj) per packet, the native RPL has a low PDR causing an increase in the number

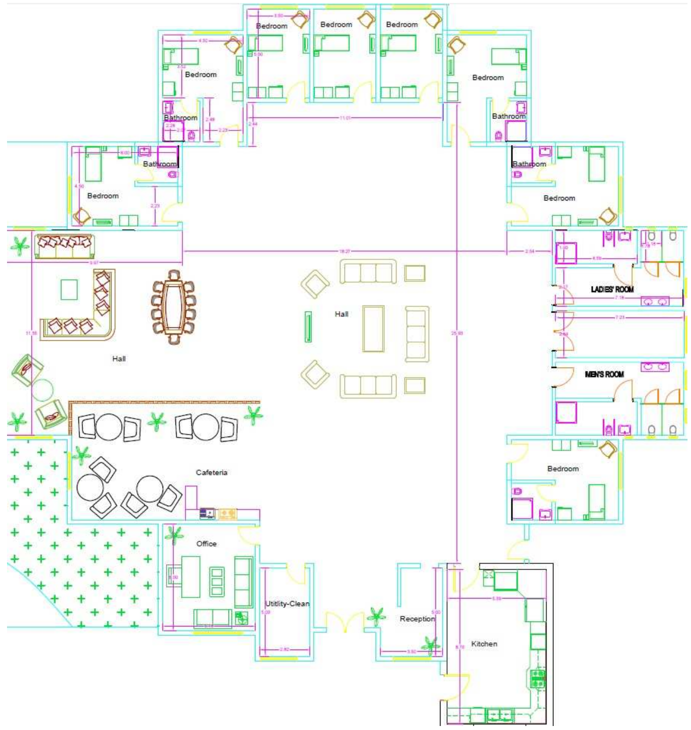

Fig. 4. A typical elderly care unit

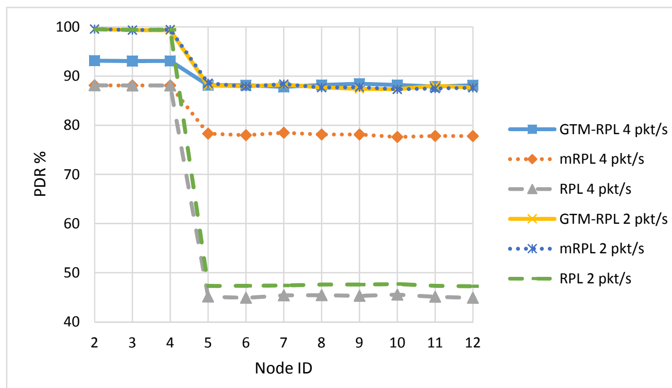

Fig. 5. PDR

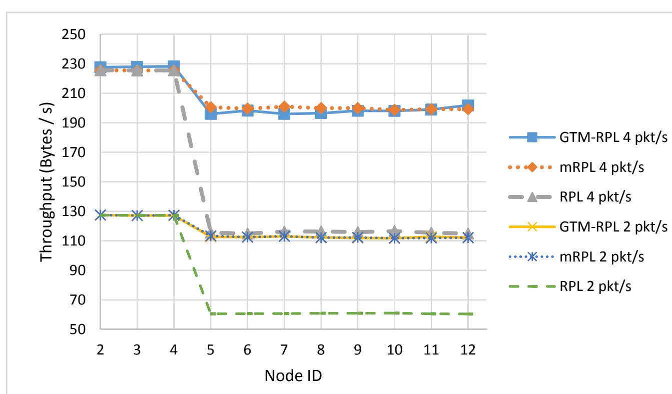

Fig. 6. Throughput

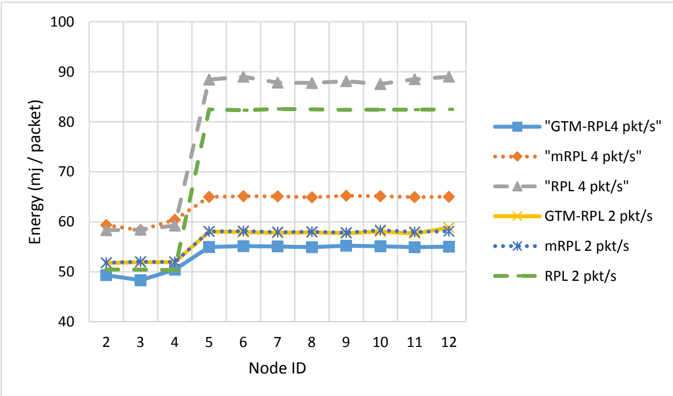

Fig. 7. Energy consumption

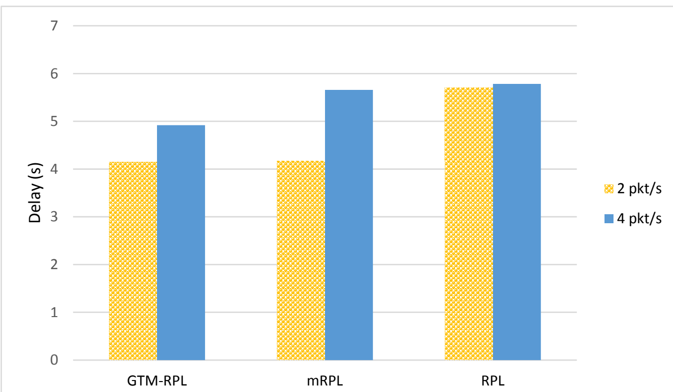

Fig. 8. End-to-end delay

of lost packets and thus a high energy consumption per successfully transmitted packets. At 2 pkt/s, mRPL and GTM-RPL achieve similar energy consumption per packet but at 4 pkt/s, GTM-RPL shows an improvement of more than 16% energy consumption for the same throughput compared to mRPL due to higher packet loss in mRPL. Although GTM-RPL aims to maximize the data transmission rate at each node, it takes into account the mobility of nodes and the noise level caused by higher transmission rates. The presence of mobility affects the value of $RSSI$ and the transmission rates of neighbouring nodes affect the value of LQI and thus LQ, both $RSSI$ and LQ are important parameters in the selection of the optimum transmission rate.

Fig 8 shows the average end-to-end delay for packets travelling from the application layer of the sending node to the application layer of the receiving node. At a transmission rate of 2 pkt/s, GTM-RPL and mRPL show similar results because the number of nodes and the frequency of transmission are not high enough to cause an increase in the LQ cost. At a transmission rate of 4 pkt/s however, GTM-RPL has 15% lower average end-to-end delay compared to mRPL. The native RPL has an average end-to-end delay of more than five seconds for both transmission rates because it is less responsive to network changes and has no efficient way of managing mobility.

B. Hospital Environmental Monitoring

In this application, one sink node and 11 sensor nodes are deployed in one of St James's hospital wards in Leeds. As shown in Fig 9, the area in the middle is not accessible leading to a different mobility limitation. Three of the sensor nodes are fixed in range of the sink node while the other eight nodes are attached to patients, equipment and staff to provide a wider sensing area and more accurate readings. The sensor nodes read a range of information including temperature, humidity and light levels and send it through the sink node to actuators in order to take an action and either fix the problem automatically (e.g. opening a window) or inform the appropriate entity, sensors also read patient data and monitor their medical condition. It is assumed that two of the patient nodes, number 5 and 6, have a high risk of emergency and thus give them a high priority of 1 while giving the rest of the nodes a normal priority of 5. Nodes with higher priority focus more on sending the data at higher rates and worry less about energy consumption compared to nodes with lower priority. This application requires high throughput because of the wide range of data and the probability of urgent incidents. For this scenario, three different transmission rates of 4, 8 and 16 pkt/s are used for testing.

The simulation results for this application are shown for three protocols, GTM-RPL, mRPL and the native RPL each at three transmission rates 4, 8 and 16 pkt/s. Fig 10 shows the PDR for each protocol using the three different settings. GTM-RPL uses an adaptive transmission rate that changes during operation and reaches a maximum of 4, 8 and 16 pkt/s depending on the configuration, while RPL and mRPL use a fixed value of 4, 8 and 16 pkt/s and do not change it during operation. At a transmission rate of 4 pkt/s, the results are relatively similar to the first scenario with GTM-RPL outperforming mRPL by around 10%. Using a transmission rate of 8 pkt/s, the effect of LQ becomes more obvious and GTM-RPL transmits at around 6.2 pkt/s for normal priority nodes and at 6.5 pkt/s for high priority nodes to avoid packet loss while mRPL and RPL send data at 8 pkt/s causing higher packet loss due to high noise and traffic congestion. It can be seen that GTM-RPL has an improvement of more than 25% in terms of PDR compared to mRPL. At a transmission rate of 16 pkt/s, GTM-RPL keeps the same transmission rates (6.2 - 6.5 pkt/s) given the same mobility model and the same network conditions. It is clear to see that mRPL and RPL nodes sending at 16 pkt/s have less than 25% PDR due to high noise and congestion.

Fig 11 shows that GTM-RPL achieves similar throughput at a transmission rate of 4 pkt/s while GTM-RPL outperforms mRPL by 10% and 50% at transmission rates of 8 and 16 pkt/s respectively. At 16 pkt/s, mRPL has lower throughput compared to the same protocol sending at 8 pkt/s. This indicates that, although increasing the transmission rate seems like the right solution to optimize throughput. Sending data at rates that are too high can deteriorate the throughput due to significantly higher levels of packet loss. The throughput at nodes 5 and 6 show slightly higher throughput than the rest of the mobile nodes showing the effect of priority on node

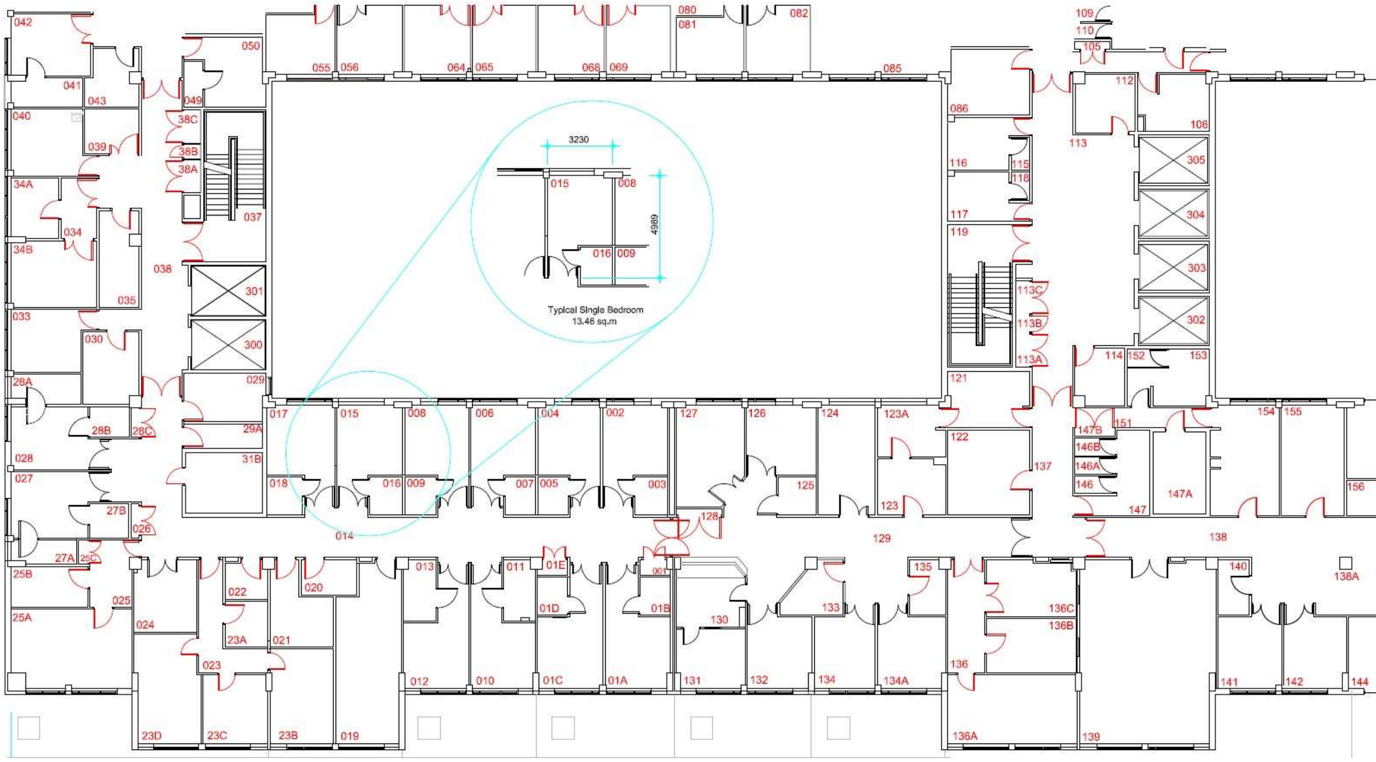

Fig. 9. Hospital environmental monitoring

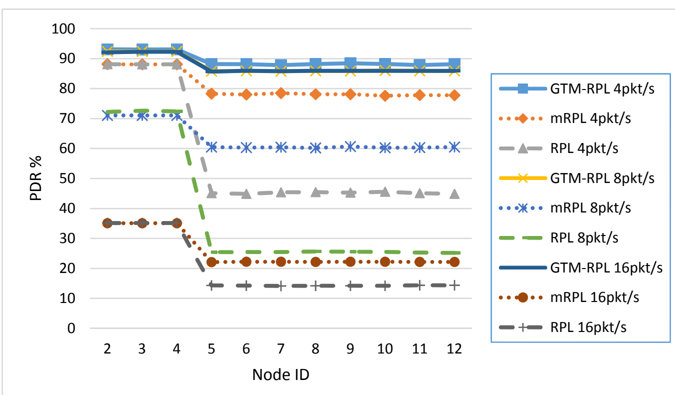

Fig. 10. PDR

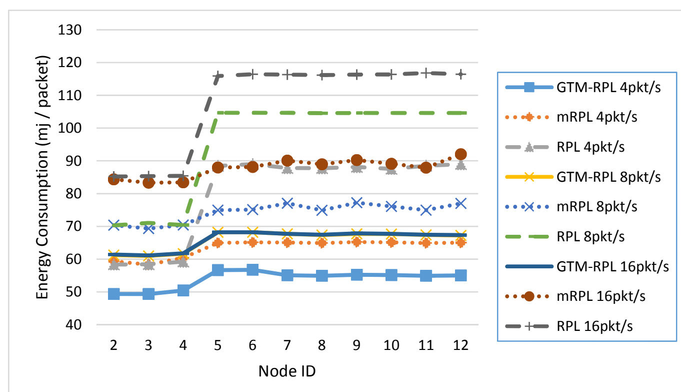

Fig. 12. Energy consumption

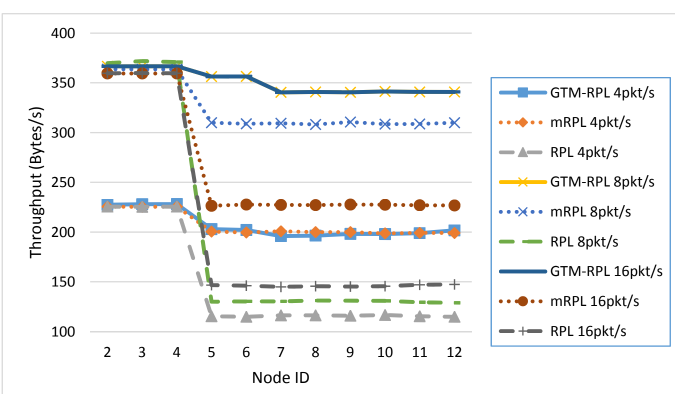

Fig. 11. Throughput

performance.

The energy consumption levels in Fig 12 show that GTM-RPL maintains relatively low energy consumption for all settings outperforming both mRPL and RPL. The native RPL has a very high energy consumption per successfully transmitted packet due to high packet loss especially for mobile nodes. GTM-RPL and mRPL on the other hand do not lack the efficiency in managing mobile nodes and thus the difference in energy consumption between static and mobile nodes is less significant.

The average end-to-end delay in Fig 13 shows the average time that a packet needs to travel from the application layer of the sending node to the application level of the destination. One of the main causes of high delay in RPL is congestion

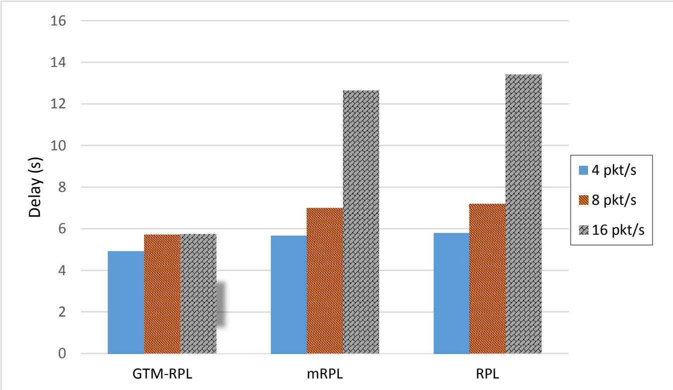

Fig. 13. End-to-end delay

[23], and it is affected by both the presence of mobility and the transmission rate of nodes. GTM-RPL avoids congestion by managing both the mobility of nodes and their transmission rate. For this reason, GTM-RPL maintains relatively low end-to-end delay at all simulated scenarios while mRPL and the native RPL have higher delay especially at increased transmission rates.

V. Conclusions

This paper provides comprehensive analysis for using RPL in a mobile environment. Game theory is used in this paper to find an optimal solution for routing depending on the application requirements. The proposed approach uses a mobility metric and a density metric that are application specific parameters, to derive the mobility cost function and the energy cost function respectively. The analyses in this paper are all based on the IEEE 805.15.4 standard and 6LoWPAN protocol stack in the presence of mobile nodes. The proposed solution is tested and evaluated using COOJA emulator over Contiki 3.0 OS, and compared against related protocols. Simulation results confirm the analysis of this paper and show that the proposed GTM-RPL outperforms existing protocols in terms of PDR, throughput, energy consumption and end-to-end delay. It provides a flexible, adaptable and expandable solution for routing in IoT applications with the presence of mobile nodes achieving higher throughput whilst consuming less energy showing more than 10% improvement compared to relevant protocols. The advantage of using GTM-RPL becomes more significant in demanding applications where simulation results show that it improves throughput by 10% - 50% showing better PDR, less energy consumption and reduced end-to-end delay. GTM-RPL offers higher performance at a lower cost taking advantage of the various parameters that contribute to the optimization game. Using $RSSI$ and LQ in addition to the improved trickle timer provides an optimized solution for routing in dynamic and mobile IoT applications.

References

[1] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, 2013.

[2] I. Korbi, M. Ben Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *Proceedings of the Third International Conference on the Network of the Future (NOF)*. IEEE, 2012, pp. 1–8.

[3] B. Tian, K. M. Hou, H. Shi, X. Liu, X. Diao, J. Li, Y. Chen, and J.-P. Chanet, "Application of modified RPL under VANET-WSN communication architecture," in *Proceedings of the Fifth International Conference on Computational and Information Sciences (ICCIS)*. IEEE, 2013, pp. 1467–1470.

[4] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of rpl under mobility," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[5] O. Gaddour, A. Koubâa, and M. Abid, "Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using rpl," *Ad Hoc Networks*, vol. 33, pp. 233–256, 2015.

[6] F. Gara, L. Ben Saad, R. Ben Ayed, and B. Tourancheau, "RPL protocol adapted for healthcare and medical applications," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2015, pp. 690–695.

[7] H. Fotouhi, D. Moreira, and M. Alves, "mrpl: Boosting mobility in the internet of things," *Ad Hoc Networks*, vol. 26, pp. 17–35, 2015.

[8] M. R. Anand and M. P. Tahiliani, "mRPL++: Smarter-hop for optimizing mobility in rpl," in *Proceedings of the IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2016, pp. 36–41.

[9] H. Kharrufa, H. Al-Kashoash, Y. Al-Nidawi, M. Q. Mosquera, and A. H. Kemp, "Dynamic rpl for multi-hop routing in IoT applications," in *13th Annual Conference onWireless On-demand Network Systems and Services (WONS)*. IEEE, 2017, pp. 100–103.

[10] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," *Internet Engineering Task Force, RFC6206*, 2011.

[11] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," 2012.

[12] O. Gnawali and P. Levis, "The etx objective function for rpl," Working Draft, IETF Secretariat, Internet-Draft draft-gnawali-roll-etxof-00, February 2010. [Online]. Available: http://www.ietf.org/internet-drafts/draft-gnawali-roll-etxof-00

[13] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Tackling mobility in low latency deterministic multihop ieee 802.15. 4e sensor network," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1412–1427, 2016.

[14] ——, "Impact of mobility on the IoT mac infrastructure: Ieee 802.15. 4e tsch and lldn platform," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 478–483.

[15] J. Montavont, D. Roth, and T. Noël, "Mobile ipv6 in internet of things: Analysis, experimentations and optimizations," *Ad Hoc Networks*, vol. 14, pp. 15–25, 2014.

[16] C. de Waal and M. Gerharz, "Bonnmotion: A mobility scenario generation and analysis tool," *Communication Systems group, Institute of Computer Science IV, University of Bonn, Germany*, 2003.

[17] C. Gomez, E. Kim, D. Kaspar, and C. Bormann, "Problem statement and requirements for ipv6 over low-power wireless personal area network (6lowpan) routing," *Internet Engineering Task Force, RFC6606*, 2012.

[18] H. Nikaidô and K. Isoda, "Note on non-cooperative convex game," *Pacific Journal of Mathematics*, vol. 5, no. 5, pp. 807–815, 1955.

[19] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, pp. 520–534, 1965.

[20] H. Kuhn and A. Tucker, "Nonlinear programming," in *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, 1951.

[21] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.

[22] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings of the 31st IEEE Conference on Local Computer Networks*. IEEE, 2006, pp. 641–648.

[23] H. Al-Kashoash, M. Hafeez, and A. Kemp, "Congestion control for 6lowpan networks: A game theoretic framework," *IEEE Internet of Things Journal*, 2017.

Harith Kharrufa is a Ph.D. student at the university of Leeds / school of Electronic and Electrical Engineering working on routing for mobile Wireless Sensor Networks and Internet of Things. Harith received his Bachelors degree in Computer and information technology engineering from the University of Mosul (2008) and received his Masters degree in computer network administration and management from the University of Portsmouth (2009). He worked as a network engineer in the Iraqi state company for Internet services (SCIS) for 4 years contributing to a number of projects to support the network infrastructure and Internet services in the city. His research interests include data routing in WSNs, MANETs, IPv6, and Internet of Things applications looking at energy efficiency and reliability in the presence of mobile nodes.

Hayder A. A. Al-Kashoash received the Ph.D. degree from the School of Electronic and Electrical Engineering, University of Leeds, Leeds, U.K. He is a Lecturer with the Technical Institute/Qurna, Southern Technical University, Basra, Iraq. His current research interests include congestion control and resource management in WSNs, 6LoWPAN, and LPWAN by utilizing game theory and optimization theory.

Dr Andrew H. Kemp is a Director of the SWJTU-Leeds Joint School, and a Senior Lecturer in the Electronic and Electrical Engineering School at the University of Leeds, UK.
Andrew received a BSc (hons) from the University of York, UK, and PhD from the University of Hull, UK. His doctoral studies investigated the use of complementary sequences in multi-functional architectures for use in CDMA systems. Technology adopted in 3G systems. He spent several years working in Africa assisting in Seismic exploration and worked at the University of Bradford as a research assistant investigating the use of Blum, Blum and Shub sequences for cryptographically secure 3G systems. He is now at the University of Leeds where he led development of wireless fieldbus systems for industrial sites and development of ranging engines in WSN chip sets. Andrew has over 100 scientific journal and conference papers and book chapters published. His research interests are in aspects of WSNs, especially related to localization, routing and also multipath propagation studies to assist system development, cross-layer optimisation and wireless broadband connection to computer networks incorporating quality of service provision. He is now contributing to the international development of Internet of Things (IoT) provision especially in the Smart Cities field. Dr Kemp is part of the highly successful School of Electronic and Electrical Engineering at the University of Leeds and a member of the School Management Board. He is a Fellow of the Higher Education Academy, a Senior Member of the Institute of Electrical and Electronic Engineering, a Member of the Institute of Engineering and Technology, and he is a Chartered Engineer.