Computer Networks 107 (2016) 339–352

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Low-power and lossy networks under mobility: A survey

Afonso Oliveira*, Teresa Vazão

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

ARTICLE INFO

Article history:
Received 30 July 2015
Revised 22 March 2016
Accepted 28 March 2016
Available online 8 April 2016

Keywords:
LLN
RPL
IoT
6LowPAN
Routing
Mobility

ABSTRACT

With the creation of the Routing Over Low power and Lossy networks (ROLL) group, work centered on the Internet of Things (IoT) has been emerging. A routing protocol for Low-power and Lossy Networks (LLNs) named the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) has been created recently, though it still has some issues, including its lack of responsiveness to mobility. This article surveys proposed mobility extensions to the RPL and analyzes how the mechanisms introduced affect the requirements for LLNs.

1. Introduction

In the Internet of Things (IoT), objects that surround us in our daily life will be connected to a network. With the emergence of these connected smart objects, there is a need to build a routing protocol that can run in resource-restrained devices and aims to reduce energy consumption because a considerable amount of nodes in an IoT scenario is battery powered. To this purpose, the Routing Over Low power and Lossy networks (ROLL) working group from Internet Engineering Task Force (IETF) was created, and they introduced the term Lossy Networks (LLNs) to to characterize these network scenarios.

To cope with the resource limitation of LLN networks, the Institute of Electrical and Electronics Engineers (IEEE) created a new Medium Access Control (MAC) protocol, IEEE 802.15.4. The limited capabilities introduced in this standard, such as the low MTU size (127 bytes) and low data rate (250 kbps) [3], put a great barrier in the implementation of IPv6-based routing protocols, as IPv6 has a minimum Maximum Transmission Unit (MTU) size of 1,280 bytes [9]. For this purpose, Low-power Wireless Personal Area Network (6LoWPAN) [24] creates an adaptation layer between the network and a data link to support IPv6 in LLNs.

In the ROLL working group, IETF studied the different application scenarios in which LLNs might be used. The simplest scenario is home automation, which aims to support our in-house daily life activities. Simple applications exist to monitor and control lightening and shutters, appliances or healthcare devices; more complex ones may be used to perform video surveillance, generate security alarms and provide overall energy optimization [5]. These applications use different sensors and actuators that may be battery operated or have quite limited resources. The majority of nodes in home networks are fixed in some part of the house. However, mobility is expected to arise mostly from healthcare devices or remote controls that are carried by people.

Similar to the home automation scenario, though applied to a much wider and more structured space, is the building automation scenario [28]. These areas of application involve mainly the control of air conditioning, elevators, lightning and shutters and the monitoring of fire sensing and security (i.e., motion detectors) devices. Such a variety of applications leads to a more diverse type of node and a higher traffic volume. Although most of the devices are fixed, some applications need to respond well to mobility.

In an industrial scenario, wireless devices are used to monitor and control devices that were not connected in the past [30]. Examples include sensors that report vibration levels on pumps, the state of a fuse or luminary and whether a man is down. Network attributes will vary greatly with the type of industry and the monitored devices; however, most application scenarios require networks of hundreds of devices that might be clustered into smaller networks around different *sinks*. Industrial applications can involve mobile sensors in containers or vehicles as well as workers carrying Personal Digital Assistants (PDAs). Thus, different types of mobility must be supported. Also, with higher speed values being introduced, the negative impact of mobility on signal quality [33] should be accounted for.

* Corresponding author. Tel.: +351916193769.
E-mail address: afonso.oliveira@tecnico.ulisboa.pt (A. Oliveira).

http://dx.doi.org/10.1016/j.comnet.2016.03.018

Finally, in an urban scenario, networks are expected to have thousands to millions of nodes that are usually separated through different *sinks* [10]. Areas of application can include sensors to measure municipal consumption (i.e., gas, water, electricity), meteorological, pollution and ambient data; actuators to control traffic and street lights also constitute an application area. Despite the network scale and diversity, currently, no mobility is foreseen for this particular type of application scenario.

Analysis of the different application scenarios has shown that the routing protocol for LLNs must be able to cope with resource limitation, scalability and Quality of Service (QoS) issues. Apart from the urban scenario, in which the nodes stand still, routing must also support low mobility in all remaining cases.

Several routing protocols have been proposed to handle these issues, and IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) was considered the *de facto* standard for routing over LLNs. The diversity of solutions leads to different surveys where the protocols are described and compared, mainly regarding their resource usage. Hence, a survey of the protocols designed for networks with low power motes is presented in [37]. Motes are sensor nodes, which are known for having limited memory and computational power. Also a comparison of RPL with previous energy-aware protocols is presented in [15].

Mobility support is a fundamental issue for the success of LLN. Traditional solutions are often too complex to be adequate for such resource-restrained and large-scale networks, and a routing protocol that can provide such support while considering these aspects will offer better performance. Although different mobility extensions have been proposed for RPL, there is no way to assess them in a unique framework.

This paper aims to compare different routing protocols for LLNs and assess the impact of mobility on the network performance when they are used.

The paper provides a structured study of RPL and the most relevant extensions that have been proposed by different authors to cope with mobility. Using one of the most used mobility models, a simulation study was conducted to assess, under the same framework, the performance of the different protocols under different mobility conditions. The goal was to identify how well they adapt to the requirements of the LLN application scenarios.

The remaining paper is structured as follows: Section 2 provides an overview of low-energy routing protocols and details the RPL protocol; Section 3 presents the mobile RPL extensions, providing an overall analysis at the end; Section 4 assesses their performance under different mobility conditions through a set of simulation studies; and Section 5 concludes the paper.

2. Routing over low power networks

Power-aware routing solutions and protocols have been presented for a long time, even before the creation of the ROLL group of IETF. Specific metrics and protocols have been proposed, especially for wireless sensor networks, with the aim of supporting nodes with energy consumption limitations. This section presents the most relevant approaches.

2.1. Routing for low energy

In 1998, Singh et al. introduced different energy-aware metrics for routing over low power [34]. In 2000, Low–Energy Adaptive Clustering Hierarchy (LEACH) [17] was proposed. LEACH is a cluster-based routing protocol that rotates the cluster head nodes to preserve the network lifetime in terms of power. Another cluster-based protocol that assigns the gateway role to a less energy-constrained node was proposed in [40]. These approaches typically try to find the minimum energy path to optimize energy usage at a node. A different perspective was followed in an energy-aware routing approach that uses sub-optimal paths occasionally to provide substantial gains [31]. The use of a probabilistic forwarding mechanism to send traffic through different routes with the same objective to avoid excessive energy consumption was also proposed [27].

Later, with the standardization of 6LoWPAN [24], new protocols were designed to support IPv6 under low-power networks. Hierarchical routing (HiLow) is a hierarchical routing protocol that aims to resolve scalability issues [21]. Based on Ad Hoc On-Demand Distance Vector (AODV) [29], which is a well-known distance-vector protocol for Mobile Ad Hoc Networks (MANETs), two protocols were proposed: 6LoWPAN Ad-Hoc Ondemand Distance Vector Routing (LOAD) [20] and Sink Routing Table over AODV (S-AODV) [6]. LOAD is a simplified version of AODV, whereas S-AODV introduces a new mechanism to deliver a better performance and less energy consumption when traffic is usually routed to a *sink node*. There is also a 6LoWPAN adaptation of Dynamic MANET On-demand (DYMO) [7], a distance-vector protocol now known as AODVv2, which is Dynamic MANET On-demand for 6LoWPAN Routing (DYMO-Low) [19].

Because none of the IETF standard routing protocols provide acceptable performance under the unique conditions that characterize LLN, the IETF ROLL group developed RPL, a distance-vector protocol designed to answer the routing requirements of this kind of network. This protocol has been defined in the recent literature [22] as the *de facto* standard for IoT.

2.2. RPL

The RPL [38] is a loop-free, distance-vector protocol designed for IPv6 LLNs. Using a proactive approach, this routing protocol builds a logical tree over any physical topology that consists of a direct acyclic graph per *sink node*, the *Destination-Oriented Directed Acyclic Graph (DODAG)*. Each DODAG defines the direct routes from every node of the network to the *sink node*. The DODAG is created using an *Objective Function (OF)* that defines the routing topology by considering the selected metrics and constraints in the building process. A simple example of a metric is the hop-count, and a simple example of a constraint is "avoid battery powered nodes."

Depending on the graph, nodes may assume different roles in the network: the *sink node* is the root of the graph and is the one that is connected to the Internet or any outer infrastructure, *router nodes* are placed in intermediate positions of the graph and are used to forward traffic to the *sink*, and finally, *leaf nodes* are located at the edges of the graph and cannot be used to forward traffic. Each node, apart from the *sink*, is configured as either a *router node* or a *leaf node*.

2.2.1. DODAG - construction and maintenance phase

RPL creates and maintains at least one DODAG per *sink node*. It computes *upward* and *downward* routes independently, according to the process described in the next sections.

Upward route. The first step of RPL consists of creating a graph that every network might use to select the route used to send data to the *sink node*, the *upward route*. The three phases of this process are illustrated in Fig. 1.

Initially, the *sink node* advertises its existence by broadcasting a *DODAG Information Object (DIO)* message to its neighbors. The neighbor nodes then process the message and make the decision of whether to join the graph. The decision of joining the graph is based on the node's *rank*: an incremental value calculated using as reference a predefined OF. Nodes with a *rank* lower or equal than the current value will not join the graph. If the node is configured

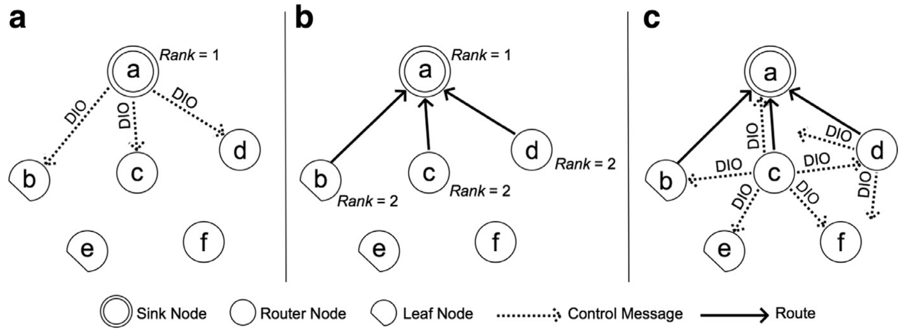


Fig. 1. Upward route construction: (a) *sink node* calculation and dissemination process; (b) router node calculation and dissemination process; (c) leaf node calculation.

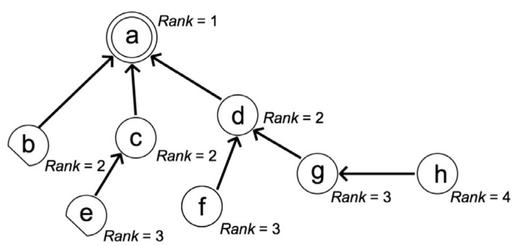


Fig. 2. Example of a DODAG with only upward routes.

as a *leaf node*, after updating its *rank*, it may join the graph without sending any information. Conversely, a node that is configured as a *router* node and decides to join the graph will continue the DODAG building process. After updating the DIO message with its own information, it broadcasts the modified message to its neighbor nodes. After this process is completed, all of the nodes in the network will have at least a routing entry to its parent that hop-by-hop lead to the *sink node*. This represents a Multipoint-to-Point (MP2P) forwarding model—also referred to as upward routing—where each node can reach the *sink*, as depicted in Fig. 2.

Another situation occurs when a new node joins an already defined network. For that purpose, the node sends a DODAG Information Solicitation (DIS) message. This message is used to proactively request graph information from its neighbors via DIO messages. After receiving the DIO messages, the new node then selects one or more parent nodes using the OF.

To maintain the DODAG, each node (except for a *leaf node*) periodically sends a DIO message to its neighbors. The interval between messages will vary according to a Trickle timer [26]. This timer dictates the interval between DIO messages by increasing it as the network becomes more stable. Upon detecting inconsistencies in the network, the timer will reset the interval to a minimum value. By using this mechanism, the frequency of DIO messages will be adapted to the network stability, and therefore, fewer messages will be sent when they are, at least theoretically, required less frequently.

Downward routes. The second step of DODAG is related to *downward route* construction and maintenance. These types of routes are needed to support Point-to-Multipoint (P2MP) communication, for instance, when the *sink node* or another entity outside the network needs to communicate with nodes in the network.

In terms of downward routing, RPL has two modes of operation: *non-storing mode* and *storing mode*. One DODAG can only be in one MOP; thus, there cannot be nodes operating in different modes in the same graph.

In the *storing mode*, RPL *router nodes* maintain a routing table with all reachable destinations. The entries of this table are refreshed upon recept of a DAO message; when the associated timer expires, the node aggregates the routing information from all of the DAO messages received and sends a new DAO message to its parent set. The process repeats until the DAO messages reach the *sink node*.

In the *non-storing mode*, RPL behaves as a source-routing protocol. The DAO messages are sent directly to the *sink* node, which records all of the routes to all of the nodes.

If downward routing is enabled, DIO messages carry a trigger for the receiving nodes to generate a DAO message. This DAO message contains the list of parent nodes of a node.

Loop avoidance and detection. Link failures or a loss of control traffic may lead to the creation of routing loops. In RPL, a loop can be caused by a node that loses connection to its parent for some reason and chooses as a new parent the child of one of its children. RPL uses loop avoidance mechanisms such as limiting how a node can move deeper in the graph (increase *rank*) within the same DODAG version.

Even with loop avoidance mechanisms, loops can still occur. RPL also has mechanisms to detect loops. One way in which RPL implements this mechanism is by using the IPv6 routing header bit. This bit will be set to indicate whether it is traveling upwards or downwards, if there is a routing anomaly that a node can detect or if the packet is being forwarded correctly.

2.2.2. DODAG - route repair phase

When inconsistencies are detected, there is a need for repair. This repair can be done either locally or on a global scope. Local repair is a mechanism for patching inconsistencies such as a loop detection or link failure. It consists of solving the detected inconsistency without restructuring the DODAG from the root. For instance, if a node loses connection to its parent, a new parent will be chosen.

Local repairs do not yield optimal solutions. Each time a local repair is executed, the DODAG will diverge from its optimal state because a global repair will eventually be needed. A global repair will restart the building process and give a new version number to the DODAG. A global repair re-optimizes the structure but has a cost in terms of network performance because the graph must be calculated anew, which will increase the control traffic in the network.

2.2.3. Data forwarding

RPL supports both MP2P and P2MP communications, as upward and downward routes are proactively built and maintained. By having these routes established, the protocol also supports Point-to-Point (P2P) communication.

When P2P communication is used in the *storing mode*, communication between two nodes is achieved by sending packets upward until a common ancestor is found and then downwards from the common ancestor to the destination node. In the *non-storing* *mode*, the packet will have to travel through the *sink node* because intermediary nodes do not record routing information.

3. RPL mobility extensions

RPL was designed to support LLNs, although there are still some open issues that need to be resolved [32]. Performance studies have shown that the protocol provides a very fast network setup with a relatively high overhead [4] and that mobility support can be greatly improved [25]. Hence, several extensions and modifications have been proposed to solve these problems. Most of them attempt to fix the lack of responsiveness or excessive overhead (due repairs) in mobile scenarios.

The next sections details the RPL extensions that aim to cope with node mobility.

3.1. DAG-based multipath routing (DMR)

DMR [18] aims to deal with frequent topology variations that arise in mobile sensor networks (MSN) by using an altered version of RPL. Their main aim is reliability of data transfer on mobile scenarios. The introduction of the link quality indication (LQI) [16] value in RPL, which was also used in more recent works, is why DMR was chosen for the survey.

To build the DODAG, DMR uses hop-count for *rank* calculation because it has been referred to as a good link-quality metric for mobile scenarios [11]. It also uses LQI [16] to determine priority among nodes with the same *rank*. The node uses the received signal strength indicator (RSSI) value received from the chip to compute the LQI.

When a node receives a DIO message, it computes the *rank* and LQI and determines whether to accept the sender node as a parent. Independent of this decision, the node stores a tuple with the relevant link information (node ID, DODAG ID, *rank* and LQI). This tuple will be recorded for DIO messages from nodes with a *rank* smaller than that of the node processing the message and will be the node's routing entry.

Packets will be forwarded first to the node with inferior *rank*, which has a greater LQI. If no node with inferior *rank* is reachable, the packet will be forwarded to the node with the same *rank* (sibling) that has a greater LQI value. If no node in the routing table is reachable, then the node solicits graph reconstruction by multicasting a DIS message directed at the *sink*.

DMR stores more routing information in each node to obtain more paths. This can be an issue because some of the typical nodes used in LLNs are known to have memory limitations. Additionally, DMR does not propose a downward routing scheme because it is the only traffic model that supports MP2P directed at a *sink*.

3.2. Modified RPL for vehicular ad hoc networks (VANETs)

In the work [25], the authors tested the RPL protocol on a VANET scenario, where the nodes are mobile. They state that RPL has slow response to topology changes, which results in suboptimal paths, loops and unreachability of some nodes. To tackle these challenges, the authors also proposed RPL modifications for mobility, one of which is the use of a fixed time DIO trigger, which was also used in subsequent works. Hence, these alterations are introduced in this survey. We refer to these alterations as MRPL-V.

The authors consider the RPL implementation and their modifications by using ETX as a metric for the OF. In terms of route maintenance, one alteration proposed by the authors is to immediately perform ETX probing upon the discovery of a new neighbor, as opposed to a periodic strategy. Another modification is that DIO and DAO messages are immediately triggered after a new parent election, ignoring the respective Trickle timer and DAO delay. Finally, the possibility of a fixed interval to trigger DIO messages is proposed, as opposed to the Trickle timer used by the standard.

For route repair, there is one alteration: when choosing a new preferred parent. a node is not able to select a previous child as the parent to avoid loops.

3.3. Mobility enhanced RPL (ME-RPL)

ME-RPL [23] is another RPL enhancement to better support mobility. The authors state that the Trickle timer is designed for static networks and that the timer alterations will be performed in all of the nodes that the mobile node passes. The authors conclude that mobile nodes should be identified and have different behaviors because although they represent a small part of the network, they have a great impact on it. In this work, nodes are separated as mobile and fixed nodes, which was also considered in subsequent works.

Another alteration in ME-RPL is in the route discovery phase. The preferred parent selection accounts for whether a node is mobile: if there is more than one node in the parent set with a similar *rank*, the fixed nodes have priority over the mobile nodes. With this, the need for a node to redefine its preferred parent due link failure or *rank* change is lower, which gives the network more stability.

The last alteration is in the route repair phase. In the standard RPL, when a node notices that it has dissociated from the network, it sends DIS messages with a fixed interval to rejoin the DAG. In ME-RPL, this interval is dynamic and depends on the number of times that a node changes its preferred parent. If a node changes its preferred parent often, it is considered to be experiencing high mobility; thus, the interval value will decrease.

3.4. Co-RPL

Co-RPL [14] enhances RPL for MSN using the corona mechanism. Its main goal is to provide better mobility support to RPL while providing QoS guarantees: mainly reliability and end-to-end delay. It also focuses on lowering energy consumption in mobile scenarios due to repair mechanisms.

In their implementation, the authors preserve backward compatibility because nodes using Co-RPL and RPL can co-exist in the same network. The Corona architecture divides the network area into coronas . Each DODAG root has its set of coronas, and they are identified by the number of hops to the respective route. For instance, a node is in corona 4 if it has a 4-hop distance to the root.

In Co-RPL DIO, messages carry a corona identifier (C_ID) that states the distance of the DIO sender node to its corona *sink* by a metric of the hop-count. The receiver of a DIO message also records an LQI value that is associated with the link between it and the sender. The preferred parent of a node will be chosen, considering the corona identifier of the candidates, and uses their link LQI value as a tiebreaker.

In terms of route maintenance, a Co-RPL node sends a DIO message immediately after receiving one, ignoring, in this case, the Trickle timer. Additionally, the Trickle timer is ignored and a DIO message is immediately sent after a new neighbor is discovered. This gives the protocol more responsiveness, but a higher overhead. It should be noted, however, that a higher responsiveness can avoid a frequent global repair and, as such, decrease the overhead. Additionally, the authors propose a fixed periodic timer for DIO messages.

For route repair, if a mobile node cannot forward a packet in the upward direction, it sends the packet backwards to any node with a higher corona level and informs its children of its temporal

Table 1
Routing protocol properties.

Protocol	Routing Protocol Properties			
	Directions	Metrics	Constraints	Messages
RPL	• Upward/ Downward	• Rank (OF)		• DIO: up route • DIS: up route • DAO: down route
DMR	− Downward	• OF=Hop-count + LQI metric		− DAO
Co-RPL		+ LQI metric		
MRPL-V		• OF=ETX		○ ETX Probe
ME-RPL			+ use mobile nodes only if needed	
MMRPL			+ avoid mobile nodes	
mRPL		+ LQI metric	+ avoid mobile nodes	

unavailability so no more packets are forwarded. The packet will continue to be forwarded until a route to a *sink* is found. By doing so, Co-RPL prevents packet dropping when some routes are temporally unavailable.

3.5. Mobility management for RLP

In [8], the authors propose alterations to the RPL protocol so it can support mobile nodes. The authors note that RPL lacks a quick detection of parent unreachability, which makes it difficult for mobile nodes to have an up-to-date *rank*. The authors propose alterations to avoid this lack of responsiveness while also aiming to avoid excessive overhead due to periodic traffic control.

Because the section in the referred article where the alterations are specified is called "Proposed Mobility Management,", the protocol implementing these alterations is henceforth referred to as MMRPL.

As in ME-RPL [23], there is a separation between fixed and mobile nodes, with one of the differences being that in this case, mobile nodes can only operate as a *leaf node*.

The alterations proposed are mainly in the route maintenance phase. When a node is aware that it has a mobile child, it changes its DIO message timer to a Reverse Trickle timer. This timer will start at a maximum configured value and decrease after each DIO message trigger. After it reaches the minimum configured value, it will restart the Reverse Trickle timer if the mobile node is still present.

To detect a loss of connectivity, mobile nodes have a configured value *Dthresh* that represents a number of DIO intervals. The sum of the *Dthresh* intervals after the last received DAO or DIO represents a timer that, when exceeded, makes the mobile node search for a new parent through the multicast of DIS messages.

In the simulation, the authors also integrate the dynamic DIS intervals from [23] and the periodic DIO messages from [25].

3.6. mRPL

In [12], the authors describe a hand-off mechanism for mobile nodes in RPL. This mechanism aims to provide the network with a lower hand-off delay and higher reliability.

The solution, mRPL, integrates a proactive hand-off mechanism (smart-HOP [13]) within RPL. The work is backwards compatible with RPL because a network with mixed nodes implemented with mRPL and RPL is possible.

The smart-HOP algorithm has two node entities, a mobile node (MN) and a serving access point (AP) node, which would be the preferred parent, and it aims to provide the MN with the best serving AP in its neighborhood across time, providing the necessary hand-off. The smart-HOP algorithm consists of two distinctive phases: *data transmission phase* and *discovery phase*.

In the *data transmission phase*, or route maintenance phase, apart from data packets and standard RPL messages, a node sends extra periodic DIS messages according to a mobility detection timer (MDT) to the preferred parent. This MDT is set according to the data generation rate; nodes with high data generation rates will send DIS messages more frequently. These DIS messages request a unicast DIO message from the serving AP containing an LQI value. This is an average value from the DIS messages received, and when it is lower than the defined threshold, a *discovery phase* is triggered. These DIO messages are responses to the periodic DIS messages and, as such, are not to be confused with the broadcast DIO messages according to the Trickle timer. There is also a connectivity timer (CT) that is reset after receiving any packet (control or data) from the preferred parent. If the timer runs out, the respective node considers its parent to be out of reach and triggers the *discovery phase*.

During the *discovery phase*, the route repair phase, a MN broadcasts bursts of DIS control messages to its neighbor nodes to obtain one DIO message per neighbor as a response with an LQI value. The bursts of DIS messages will periodically repeat, with an interval time, until a node with an LQI value above the defined threshold is found. To prevent the DIO control message responses from becoming congested, the responses are triggered by a timer with a random property for collision avoidance that gives priority to nodes with higher LQI values: the reply timer (RT).

Whenever there is a serving AP hand-off (a MN changes its preferred parent), the network is not to lose its stability. The Trickle timer is not reset when a MN changes its preferred parent. Additionally, to avoid loops and similar to [25], a node cannot change its parent to one of its children.

3.7. Synthesis and discussion

This section summarizes and compares RPL and its extensions. After an initial overview of the main properties of the routing protocols, the mechanisms used in distinct phases (route discovery, route maintenance and route repair) are analyzed.

Routing protocol properties. Table 1 summarizes the most important properties of RPL and each of its extensions. RPL is defined as the standard, and for each of its variants, the additions and removal of properties are highlighted.

Table 2
Route discovery mechanisms.

Protocol	Route Discovery			
	Traffic type	Type	Trigger	Dissemination
RPL	• Control traffic	• Proactive	• DIO: trickle • DIS: n.d. • DAO: event (received DAO/DIO)	• DIO: P2MP (downward) • DIS: broadcast • DAO: MP2P (upward)
DMR				
Co-RPL			• DIO: periodic (fixed)	
MRPL-V			• DIO: periodic (fixed)	
ME-RPL				
MMRPL				
mRPL				

A bullet point defines an item/behavior in the RPL protocol and an alteration to a previously defined item/behavior in the protocol extensions. A white bullet states alterations to items/behaviors that were not mentioned above because they are not part of the RPL standard but are used with the protocol (i.e., ETX probe). A plus sign is used when the extension adds an item/behavior to the protocol and a minus when it removes a previously defined one. An empty cell means that no alterations have been made to the standard.

The routing properties provide an overview of the working mechanism of the protocols. Four properties have been considered in this analysis: the directions of the routes that have been discovered (upward, downward), the metrics and constraints that serves as bases for path calculation and the routing messages used.

RPL uses a proactive routing strategy to support both upward and downward routes. The routes should be immediately ready when a packet is sent, which is a positive aspect due to the time-critical traffic present in LLNs. However, the overhead that is required to maintain the routes is a disadvantage to consider in that most of the nodes should have energy consumption restrictions because they are battery operated. Additionally, P2P communications are indirectly supported through the integration of upward and downward routing. Hence, non-optimal routes might be used, as two nodes can be near each other but packets between the two are forced to go up and down the architecture [39]. No specific metric and constraints have been defined, but the paths are calculated and ranked according to an OF that combines them. Therefore, the most adequate ones might be selected without compromising standard compliance, which is a positive aspect. Three different messages are used to support the protocol operation.

Apart from DMR, there are no major differences between RPL and its variations, as most of the specificities of the protocol extensions are focused on the definition of metrics and constraints or additional QoS support mechanisms.

DMR only uses downward routes, and as a consequence, no DAO messages are used. This fact makes the protocol unusable for the stated requirements. However, the method used to deal with link state variations is sufficiently interesting to justify its study.

DMR, Co-RPL and mRPL use also an LQI as a metric associated with neighbor links. This allows *router nodes* to choose preferred parents with better link quality, thereby providing an extra metric in addition to the OF. This has the disadvantage of bigger network table entries but also offers advantages. With an extra LQI value, it is possible to choose the best parent within a set of nodes with the same *rank*, yielding a better path quality overall. Additionally, links with the best quality are less likely to be broken, even with mobility involved, as it gives a hint of the nodes' proximity. MRPL-V is designed to be used with an ETX metric.

ME-RPL, MMRPL and mRPL consider only some nodes to be mobile to reduce the overhead that is required to detect link failures more often in all of the networks. Nodes are marked as fixed or mobile and are treated accordingly. In terms of routing constraints, MMRPL and mRPL consider mobile nodes to be *leaf nodes* because mobile nodes are not allowed to have children. Conversely, ME-RPL treats mobile nodes as *router nodes* but gives them less priority when a node is selecting a parent. Mobile nodes are chosen only as fail-safe if no fixed nodes with similar *rank* are available. When mobile nodes are referred to in the LLN requirements, it is advised that they be configured as *leaf nodes*. However, this last behavior can be good and fail-safe in sparse networks, as mobility in LLNs is reduced.

Route discovery. The analysis of the route discovery phase comprises both the type of messages and the mechanisms used by the protocols. Hence, four main properties are studied: traffic type (control or data), type of discovery (reactive or proactive), trigger mechanisms (event based, periodic or with a single timer) and how the messages are disseminated. The results are summarized in Table 2.

RPL uses three type of routing messages to support route discovery. In order to have updated routes without a significant overhead, a periodical Trickle timer is used to trigger upward route discovery . Performance studies have shown the advantage of this mechanism [35]. For nodes outside the DODAG, i.e., a new node in the network, upward routes can also be solicited. Downward routes are constructed after a node joins the network.

Regarding RPL mobility extensions, a few alterations are proposed to the route discovery procedures.

Co-RPL and MRPL-V both use fixed timers for DIO message propagation instead of the Trickle timer. Depending on the interval value chosen, these alterations can create a faster network deployment, but the main reason for this alteration is the routing maintenance phase.

Route maintenance. To keep the routing tables up to date, a protocol must have its routes updated frequently. The questions are how and how frequently. For this, the route maintenance analysis will account for the traffic type that is used to perform route maintenance (control or data), the timed and event triggers for the control traffic and how the messages are disseminated. The separation is established between timed and event triggers because the first

Table 3
Route maintenance mechanisms.

Protocol	Route Maintenance			
	Traffic type	Timed Triggers	Event Triggers	Dissemination
RPL	• Control traffic	• DIO: periodic (trickle)	• DAO: event (received DAO)	• DIO: P2MP (downward) • DAO: MP2P (upward) •Unicast DIO: event (DIS received)
DMR				
Co-RPL		• DIO: periodic (fixed)	✚ DIO: event (inconsistency detected)	
MRPL-V		• DIO: periodic (fixed)	✚ DIO/DAO: event (new parent) ○ ETX Probe: event (new neighbor)	
ME-RPL				
MMRPL		✚ DIO: periodic (reverse trickle) ✚ DIS: timer dthresh		
mRPL	✚ Data traffic	✚ DIS: periodic MDT ✚ DIS: timer CT		•Unicast DIO: event (N DIS received)

occur in a predictable fashion whereas the latter may or may not occur, depending on factors such as the network dynamics. The results are summarized in Table 3.

Concerning RPL, route maintenance shares the DIO and DAO messages and associated mechanisms with route discovery. Most of the changes proposed by the mobility extensions are related to the type of trigger that is used.

Co-RPL and MRPL-V introduce a fixed interval between the DIO messages, replacing the trickle timer. Although in [25], some fixed interval values for the DIO messages have proven to yield a better network performance compared to the use of a trickle timer, this network performance will depend more on the chosen configuration. When deploying large networks, such as in an urban scenario, pre-configuration should be avoided, and the more the protocol can adapt to the network, the better. This fixed interval solution is better used in small and predictable networks.

Co-RPL and MRPL-V also have triggers to the DIO messages that ignore the respective timers. Using the Trickle timer, the interval will increase when an inconsistency is detected. In Co-RPL, a DIO message is immediately sent after a detected inconsistency. This improves the responsiveness of the protocol at the cost of some extra control traffic.

In MRPL-V DAO and DIO, messages are sent immediately after a new parent is chosen; neighbors know instantly of its new parent choice and update their routing tables accordingly at, again, the cost of some extra traffic. This mechanism is good for downward routing, which tends to be a topic poorly covered in these alterations. Although most of the traffic is destined to the *sink*, the LLN requirements state that a considerable amount of it should be acknowledged. This is impossible without up-to-date downward routing entries. MRPL-V also sends immediate ETX probes when new neighbors are detected, which helps maintain an up-to-date metric value, which is important in dynamic networks.

In MMRPL, mobile nodes are identified in the DAO message. Parents of the mobile nodes are those that alter their behavior. When a node has a mobile child, it will use a reverse Trickle timer to time DIO messages. This timer starts with an interval value and decreases it after each DIO message is sent. The authors justified the use of this timer with the reasoning that when a mobile node connects with another node, it is expected that it remains connected for a long time, which might not be true depending on the mobility pattern, the mobile node velocity and the parent's reach. There is also a *Dthresh* timer for detecting a lost connection between nodes.

In mRPL, a more complex approach is used. In this approach, data packets are used to time DIS and DIO messages. A timer named MDT depends on the data generation rate. The idea is to send a DIS message after a number of packets are sent to receive a DIO replay with an LQI value. With this timer, nodes that generate more traffic have higher responsiveness; however, nodes with time-critical event traffic (i.e., alarms) should also have mechanisms to keep the connection information up-to-date. There is also a CT that is restarted upon any communication received and that triggers a route repair mechanism.

Route repair. Table 4 summarizes what triggers repairs in the protocols and how local repairs are processed. In terms of the messages used in the process, the table also summarizes their triggers and how they are disseminated. Global repairs were not considered because they consist primarily of the reconstruction of the DODAG, and no alterations are proposed.

Standard RPL detects broken links by using the standard IPv6 neighbor detection and has loop detection mechanisms to detect when a repair is needed.

Some extra mechanisms to detect or predict link failures besides the RPL standard or common RPL implementations are used in MMRPL and mRPL. In mRPL, not only the detection of broken links through extra timers or absent responses to some control messages but also broken links are predicted using an LQI value. This triggers a repair when links are not broken yet, thereby avoiding the extra loss of packets that occur while the broken link is not detected.

When repairing a route, a RPL *router/leaf node* can use DIS messages to query the neighbor nodes for DIO messages, which are

Table 4
Route repair mechanisms.

Protocol	Route Repair (Local)			
	Repair Trigger	Mechanism	Message Triggers	Dissemination
RPL	• Broken link (IPv6 ND) • Loop detected	• Look for new parent: send a DIS message to require a DIO message	• DIO: periodic (trickle) • U. DIO: event (DIS received) • DIS: periodic (fixed)	• DIO: P2MP • U. DIO: P2P • DIS: broadcast
DMR		✚ Traffic routed through siblings		
Co-RPL		✚ Traffic routed through children		
MRPL-V				
ME-RPL			• DIS: periodic (dynamic)	
MMRPL	✚ *Dthresh* timer expired			
mRPL	✚ No DIO reply from unicast DIS ✚ CT expired ✚ Low LQI value	• smart-HOP hand-off	✚ DIO: timer (Collision Avoidance) RT ✚ DIS: periodic (burst)	

required for route reconstruction. This use of the interval between the messages is not standardized, but in ContikiRPL [36] (which is the most commonly used implementation), a fixed interval is used. In ME-RPL, a dynamic interval value is used; the more a node changes its preferred parent, the shorter the interval is between DIS messages.

On the other end, mRPL sends bursts of DIS messages that will further be answered by a unicast DIO. The replies are filtered and prioritized depending on the LQI values obtained from the burst of DIS messages to avoid collisions on the DIO reply. The burst of DIS messages will cause a bigger overhead; however, this provides a node with more measures to acquire a more accurate LQI value. This is important because the value is used to choose a more reliable node, which can decrease the need for new repair.

When routing is not possible through the preferred parent, DMR and Co-RPL have a fail-safe option. DMR routes through the siblings (nodes with the same *rank*), and Co-RPL sends the traffic downwards, which leads to good coping under the corona scenario, where nodes can easily change the DODAG they belong to, possibly sending the message to another *sink* when one is unreachable. Additionally, children are informed that the parent cannot route traffic and will find an alternative route. These mechanisms are good for avoiding traffic losses and avoiding re-transmissions of traffic that memory-limited nodes may not even be able to store for further re-transmission.

Overall analysis. The results of the previous sections showed that the majority of routing protocols share most of the characteristics of RPL, and the differences are mainly related to the property of the scenario each one of them favors most. This is expected, as they are extensions of the standard.

Overall, the solutions tend to have some common strategies. ME-RPL, MMRPL and mRPL consider only some nodes to be mobile (partial mobility) as opposed to a scenario in which there is no node distinction (overall mobility), and the solutions adopt more aggressive and less global measures because the measures are only applied to some nodes. These solutions that identify some nodes as mobile have the common feature of avoiding mobile nodes as parents. MMRPL and mRPL both consider mobile nodes to be *leaf nodes*. ME-RPL gives priority to fixed nodes when a node is selecting a new parent, thus making the possibility of a mobile node parent merely fail-safe.

Control message periodicity is another common alteration. Co-RPL and MRPL-V use a fixed timer for DIO periodicity, as opposed to the standardized Trickle timer. There can be a fixed DIO or a Trickle DIO when accounting for the overall mobility. For the scenario where some nodes are considered mobile, MMRPL uses a reverse Trickle timer on a mobile node's parent to trigger DIO messages.

For DIS periodicity, in partial-mobility solutions, ME-RPL implements the dynamic DIS and mRPL implements a burst DIS in the route repair phase. mRPL also adds a periodic DIS in the route maintenance phase to have the same average link quality measure.

Event message triggers that ignore the periodic timers are also common alterations in overall mobility solutions. Co-RPL emits a DIO message when an inconsistency is detected, and MRPL-V emits a DIO and DAO when a new parent is selected. It also sends a new ETX probe when a new neighbor is detected. These event message triggers share the advantage of higher responsiveness and the disadvantage of a higher overhead.

Most partial-mobility solutions introduce extra broken link detection in mobile nodes. One implementation type is a connection timer that sends a DIS message to the connected node, resetting after any communication type, such as the *Dthresh* timer from MMRPL or the CT timer from mRPL. Anther mechanism is the low link quality detection from mRPL.

In terms of route repair, but in overall mobility solutions, some protocols implement some fail-safe operations if no new parent is available. DMR routes the traffic among siblings, and Co-RPL routes traffic downwards through its children until a new parent is found to avoid packet drops in that period of time.

4. Simulation

To assess and compare RPL with the mobility extensions, a set of simulation studies were conducted. Extensions to the RPL protocol were developed on top of the ContikiRPL implementation, which runs in the Contiki [1] operating system. The simulations were implemented using COOJA simulator [2], as it is the simulator used to simulate networks with motes running Contiki. The next section details the simulation studies.

4.1. Simulation goals

The main goal of the simulation studies is to study the impact of mobility in RPL and its extensions. To accomplish this goal, three different sets of simulations were carried out, aiming to assess the impact of the following:

- Set A: the density of the nodes.
- Set B: the number of mobile nodes.
- Set C: the network size in a low-mobility scenario.

4.2. Metrics

To assess RPL and its extensions, three different groups of metrics were considered.

The first group is used to study how each one of the protocols copes with the requirement of simplicity. It comprises two different metrics: Central Processing Unit (CPU) and radio usage. To measure them, Contiki's *powertrace* app is used. For each node, this app reports the amount of time used by the CPU and by the radio. These values are then divided by the total node lifetime, as follows:

$$CPU\ Usage = \frac{CPU\_time}{node\_time}$$

$$Radio\ Usage = \frac{radio\_time}{node\_time}$$

where $node\_time = CPU\_time + idle\_time$.

The second group of metrics is used to study the routing protocol behavior, and they are especially useful for addressing scalability issues, comprising two metrics: the control traffic and data traffic ratio. This data traffic ratio aims to estimate the average number of hops of each request. Control and data traffic are measured by registering captures of the transmission (and only the transmission) of all the nodes. Using the Wireshark tool *tshark*, all the captures are analyzed and filtered by User Datagram Protocol (UDP) (data) and Internet Control Message Protocol (ICMPv6) (control). The displayed values are the sum of control traffic volume and a ratio of the data traffic volume used by all of the nodes. These metrics are defined as follows:

$$ControlTraffic = \sum size(ctr\_pkt\_txd)$$

$$DataTrafficRatio = \frac{\sum size(data\_msg\_txd\_source)}{\sum size(data\_msg\_txd\_node)}$$

where *ctr_msg_txd* represents a routing message that was sent, *data_msg_txd_source* represents a data message generated at a source node, *data_msg_txd_node* represents a data message sent by any node, and *size* represents the function returning the message length.

Finally, the last group of metrics measures the performance of data transmission and is used to measure the ability to provide QoS. Both Packet Delivery Ratio (PDR) and delay were used because they are the two most commonly used metrics for CBR traffic assessment. PDR is the ratio between the received and sent messages, and the delay metric represents the time elapsed since the message is transmitted by the source node until it reaches the destination. It is measured by registering the simulation times of when a message was sent and received. For this, each message carries an identifier in its payload. Both metrics are given as follows:

$$PDR = \frac{\sum data\_msg\_rxd\_sink}{\sum data\_msg\_txd\_source}$$

$$Delay = T_{rxd\_sink} - T_{txd\_source}$$

where T_{rxd_sink} is the reception time of the data message at the *sink node*, T_{txd_source} is the transmission time of the data message at the source node, and *data_msg_rxd_sink* indicates whether the data message was received at the *sink node*.

4.3. Simulation scenario

COOJA was used to simulate Sky mote nodes. The simulator was configured to use Unit Disk Graph Medium (UDGM) as a radio medium with a 50 m transmission range, 70 m interference range and 70% success ratio of receiving packets at the edges.

A simple topology with one *sink* at the center and several nodes that build rings around that *sink node* was used. The simulator was configured to select the number of rings (n_r) and the distance between two adjacent rings (r). In our case, the first ring has six nodes, and each successive ring has twice the number of nodes of the previous ring. In the simulations presented for set A and B, three rings are used, which makes for a total of 42 routing nodes plus one *sink node*. To avoid a perfect topology, when generating the topology, nodes are given a 10 m radius of error when being placed.

The considered traffic pattern is Constant Bit-Rate (CBR) traffic consisting of small UDP messages (40 bytes) sent every 20 s from all the nodes to the *sink node*.

The mobility model used is Random Way Point (RWP). This model was chosen due to its simplicity and because it is one of the most commonly used models in this type of study. It was also used to assess some of the protocols under study in this paper, namely CO-RPL [14] and DMR [18]. Two speed values are considered. One aims to simulate the walking speed of a person (1.4 m/s), and the other aims to simulate a vehicle indoors or a machine (5.5 m/s). In the simulations presented, 20% of the mobile nodes are considered indoor vehicles/machines, and 80% are considered people. The pause times vary from 2 to 7 minutes, with a uniform distribution.

The simulations were performed using RPL and a set of mobility extensions. Apart from RPL, none of the extensions were available in the simulator; therefore, they were developed from scratch. From the set of protocols that have been described before, the ones that require the use of LQI and RSSI values were not considered because it is not possible to read these values from the simulator. Hence, the comparison comprises the following protocols: RPL, MRPL-V, MMRPL, ME-RPL and CO-RPL. Due to the lack of LQI information available, in our simulation, CO-RPL cannot take advantage of this measure during parent selection.

Table 5 summarizes the simulation parameters. The N/A value means the parameter is not applicable to the scenario.

4.4. Results

Set A. The first set of results illustrates how the different routing protocols behave when the network density is modified and the nodes stand still. Figs. 3, 4 and 5 depict the results of the three group of metrics.

The analysis of Fig. 3 shows that with lower density values, there is a lower consumption of resources (CPU and Radio). Due to the smaller number of neighbors of each node, less information is received, and therefore, less information needs to be processed. However, two different sets of protocols can be identified: Co-RPL and MRPL-V. They use more resources than the other three protocols (RPL, ME-RPL and MMRPL), which exhibit similar behaviors.

Regarding the impact on routing protocol, the results depicted in Fig. 4 demonstrate two different findings. First, in a sparse network, routes tend to be longer, and the data traffic ratio increase

Table 5
Simulation parameters.

Group	Parameter	Set A	Set B	Set C
General Properties	Simulator	COOJA		
	Mote Type	Sky Mote		
	Simulation Time	1h		
Physical Medium	Radio Medium	UDGM with 70% Rx ratio		
	Transmission Range	50m		
	Interference Range	70m		
Network Topology	Number of rings (n_r)	3		[1, 2, 3]
	Number of nodes	43		[7, 19, 43]
	Ring distance (r)	[20, 30, 40]	40	
Data traffic	Traffic Type	CBR/UDP		
	Traffic Periodicity	20s		
	Packet Size	around 40 bytes		
Mobility Properties	Mobile Nodes	0%	[0, 10, 20, 50, 70, 100]%	10%
	Mobility Model	N/A	RWP	
	Node Speed	N/A	1.4 or 5.5 m/s	
	Slower Nodes	N/A	80%	
	Pause Time	N/A	2 to 7 minutes (u.d.)	

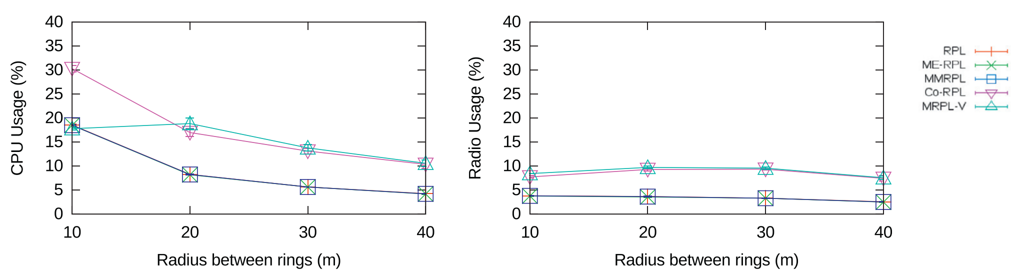


Fig. 3. Set A - Density variation: impact on resource usage.

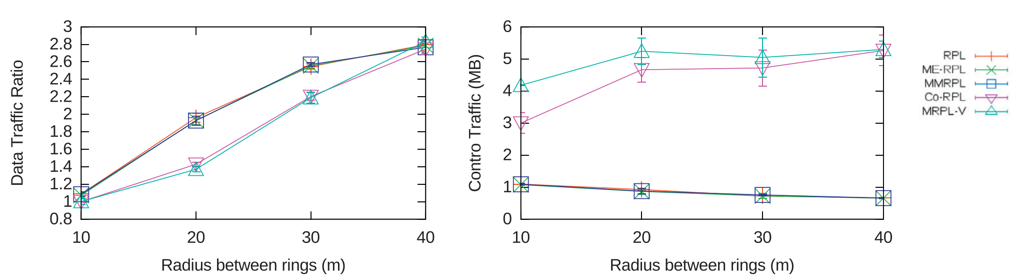


Fig. 4. Set A - Density variation: impact on routing protocol.

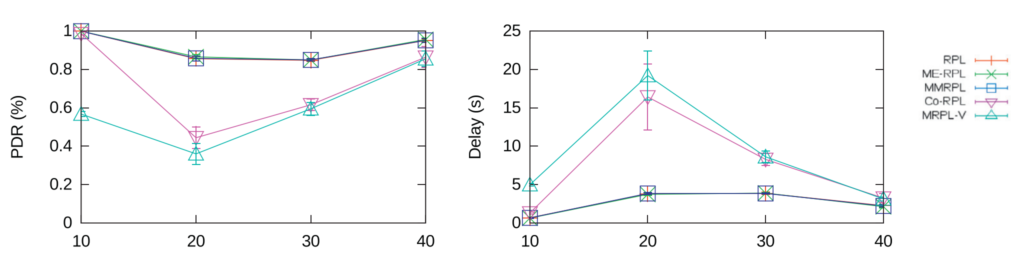


Fig. 5. Set A - Density variation: impact on data traffic.

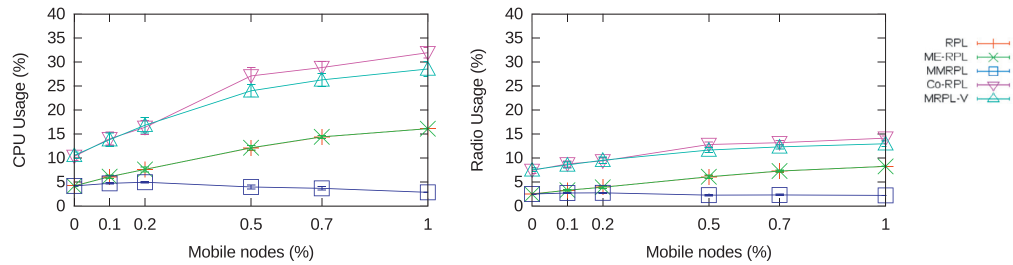


Fig. 6. Set B - Mobility variation: impact on resource usage.

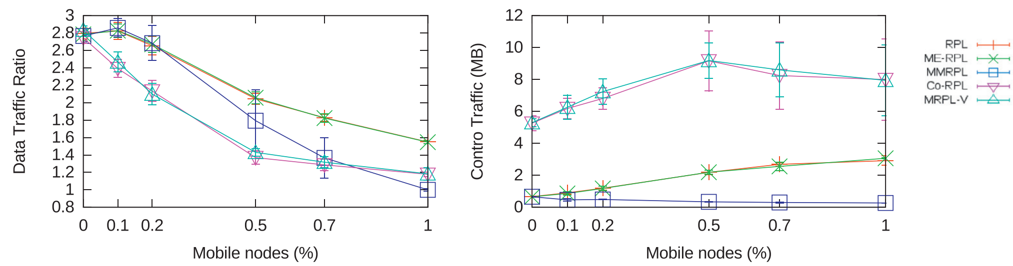


Fig. 7. Set B - Mobility variation: impact on routing protocol.

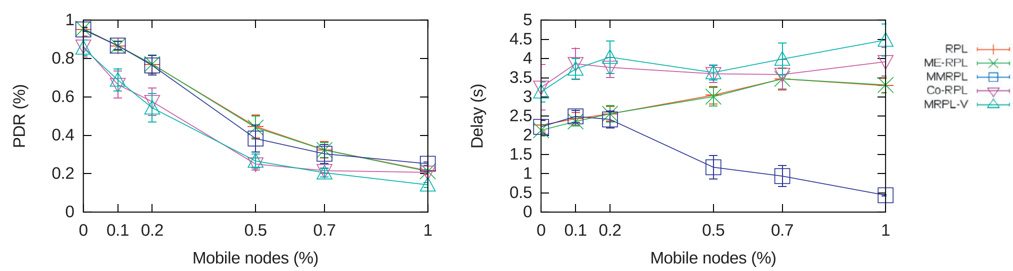


Fig. 8. Set B - Mobility variation: impact on data traffic.

because for the same amount of data traffic that is generated by a source node, more intermediate nodes are used to re-transmit it. The results are more or less similar in all of the protocols, meaning that no significant differences are found among them regarding route length. Considering the control traffic that is generated by the protocols, Co-RPL and MRPL-V use considerably more control traffic than the others. This is due to a fixed DIO interval policy that is expected to behave worse than the Trickle timer in static scenarios. Additionally, Co-RPL and MRPL-V generate control traffic based on the neighbor behavior, which make them use much more traffic in high-density scenarios. This fact also justifies the discrepancy of these two protocols in Fig. 3.

Once again, from the analysis of data traffic performance depicted in Fig. 5, it is possible to conclude that ME-RPL and MMRPL outperform the other protocols by exhibiting good PDR and delay values in all networks.

Concerning Co-RPL and MRPL-V, it can be observed that the excessive use of control traffic by Co-RPL and MRPL-V has a negative impact on PDR and Delay for networks with distance values ranging from 20 m to 40 m. However, this overhead tend to be less significant in sparser networks, which is why the PDR increases in a consistent manner when the network tends to be sparser. In the sparser scenario, the delay is lower, probably due to lower resource usage by each node, as packets can be processed and re-transmitted faster.

The difference in behavior from the 10 m to the 20 m distance scenario is due to the fact that in the 10 m scenario, all of the nodes reach the *sink node*. No routing is necessary, thereby increasing the probability of faster reception.

From this analysis, it can be observed that more verbose protocols with higher amounts of control traffic easily have a negative impact in denser scenarios, as 802.15.4 networks are known to have limited bandwidth.

Set B. The second set of results are used to analyze the impact on the percentage of mobile nodes on the behavior of the protocols, considering the same metrics. For this set of tests, the sparser scenario was chosen because it leads to a more intensive usage of the routing protocols, as the routes tend to have more hops. The plots that summarize this set are represented in Figs. 6, 7 and 8.

For simplicity, it can be observed from Fig. 6 from the CPU and radio usage plots that overall, nodes tend to use more resources in more mobile scenarios, with Co-RPL and MRPL-V using more resources in the static scenario. However, MMRPL has the opposite effect because it considers mobile nodes to be *leaf nodes*, which do not broadcast DIO messages. Fewer DIO messages are sent, which has an impact on CPU usage due to the processing of the reception of these messages (considerably higher than the processing of DAO and DIS messages).

From both plots of Fig. 7, three distinct behaviors can be observed. The MMRPL data traffic ratio starts by accompanying ME-RPL and RPL, and with the introduction of mobility, this value quickly goes to 1 as mobile nodes do not perform routing operations in this protocol.

A difference can be observed in the lines of ME-RPL/RPL and Co-RPL/MRPL-V in the data traffic ratio plot. This could be because Co-RPL/MRPL-V have more up-to-date routes and because the data traffic does not require as many hops. Following the control traffic

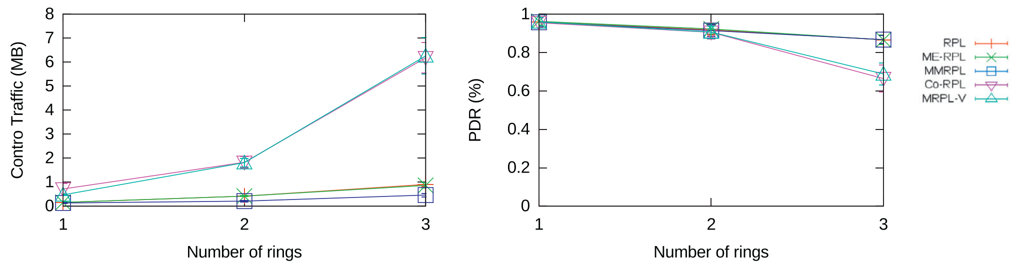


Fig. 9. Set C: Network size variation: impact of control traffic on PDR.

plot, it is interesting to observe that Co-RPL/MRPL-V starts using less control traffic from the 50% mobility scenario, whereas ME-RPL/RPL continues to use more control traffic with increased mobility. This may be due to a higher need for global repairs.

As can be observed in Fig. 8, overall, the PDR reduces with the introduction of mobility. However, the curves have different shapes. Co-RPL and MRPL-V suffer a variation in the more mobile scenario, and MRPL-V uses event-based traffic on new neighbors, which are quickly lost resulting, perhaps, in control messages that are not so efficient.

Although MMRPL does not use routing in the more mobile scenario, it slightly outperforms the other protocols in terms of PDR. This sub-performance of the other protocols can be due to two reasons: first, the routes are never properly up to date, which makes nodes in range of the *sink node* send traffic in the wrong direction, and second, the high usage of the spectrum due to the excessive use of control messages ends up having a huge negative impact on the PDR.

Set C. The final simulation set aims to study the impact of the network's size on the network performance. For this, we selected just two metrics (control traffic and the PDR) that enable us to assess the trade-off between keeping routing information up to date and delivering data. Note that in this test, we used fewer mobile nodes to represent a scenario that was as close as possible to a small LLN. The results for control traffic and PDR are presented in Fig. 9.

The analysis of the plots confirms that the event-based traffic from MRPL-V and Co-RPL makes these two protocols use more traffic. Although true for all of the networks tested, the results are more relevant in larger or denser networks.

Nevertheless, when the network is small, this trend does not have a direct impact on the PDR, as there are no significant differences among the different protocols. In the larger network, the higher usage of control traffic is felt in the PDR that decreases significantly when compared to other protocols. It can be seen that an excessive use of control messages has a considerably negative impact on the performance of data traffic.

5. Conclusion

This paper surveys routing protocols for LLNs, considering RPL and the most relevant mobility extensions that have been proposed. For each protocol, it analyses the different phases and identifies the different mechanisms and messages that are introduced by the extensions. As is expected, some mechanisms introduce trade-offs mainly in simplicity and scalability against QoS and mobility awareness.

To assess these trade-offs, RPL and some of its extensions were simulated in COOJA under different conditions. Based on the results, we can conclude that the protocols can be divided into two different groups. The first group comprises the protocols that, although they are more responsive to mobility, end up achieving the worst performance due to the excessive amount of control traffic that they require in order to operate. The second group comprises the less responsive protocols with fewer control traffic that deliver better performance. This means that the control traffic needed to keep up-to-date routing tables have a big negative impact. LLNs are networks that operate at a low data rate, with limitations in the usage of the spectrum.

More spectrum-efficient protocols are required to contemplate possible future scenarios for LLNs that require more mobile nodes.

For future work, different scenarios should be considered with more than one *sink node*, with different mobility models and traffic patterns for imitating the applications scenarios for LLNs. Additionally, experiments with real motes for testing the behavior of LQI-based mechanisms should be conducted.

Acknowledgments

This work was supported by national funds through Fundaçãopara a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

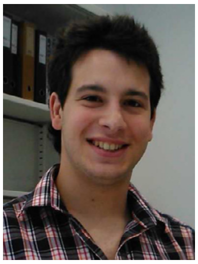
Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.comnet.2016.03.018.

References

[1] Contiki: the open source operating system for the internet of things, (http://www.contiki-os.org/). [Online; accessed 11-January-2016].
[2] The cooja network simulator, (http://www.contiki-os.org/). [Online; accessed 11-January-2016].
[3] Ieee standard for local and metropolitan area networks–part 15.4: low-rate wireless personal area networks (lr-wpans), IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006) (2011) 1–314, doi:10.1109/IEEESTD.2011.6012487.
[4] N. Accettura, L. Grieco, G. Boggia, P. Camarda, Performance analysis of the RPL routing protocol, in: Mechatronics (ICM), 2011 IEEE International Conference on, 2011, pp. 767–772, doi:10.1109/ICMECH.2011.5971218.
[5] A. Brandt, J. Buron, G. Porcu, Home automation routing requirements in low-power and lossy networks, 2010, (RFC 5826 (Informational)), pp. 1-17.
[6] Z. Cao, G. Lu, S-AODV: sink routing table over AODV routing protocol for 6lowpan, in: Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on, 2, IEEE, 2010, pp. 340–343.
[7] I. Chakeres, C. Perkins, Dynamic manet on-demand (dymo) routing, draft-ietf-manet-dymo-14 (work in progress)(2008).
[8] C. Cobarzan, J. Montavont, T. Noel, Analysis and performance evaluation of rpl under mobility, in: Computers and Communication (ISCC), 2014 IEEE Symposium on, 2014, pp. 1–6, doi:10.1109/ISCC.2014.6912471.
[9] S. Deering, R. Hinden, RFC 2460 Internet protocol, version 6 (IPv6) specification, Internet Engineering Task Force, 1998.
[10] M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing requirements for urban low-power and lossy networks, 2009, (RFC 5548 (Informational)).
[11] R. Draves, J. Padhye, B. Zill, Comparison of routing metrics for static multi-hop wireless networks, SIGCOMM Comput. Commun. Rev. 34 (4) (2004) 133–144, doi:10.1145/1030194.1015483.
[12] H. Fotouhi, D. Moreira, M. Alves, MRPL: boosting mobility in the internet of things, Ad Hoc Netw. 26 (2015) 17–35.

[13] H. Fotouhi, M. Zúñiga, M. Alves, A. Koubâa, P. Marrón, Smart-hop: a reliable handoff mechanism for mobile wireless sensor networks, in: Wireless Sensor Networks, Springer, Berlin Heidelberg, 2012, pp. 131–146.
[14] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, M. Abid, Co-RPL: RPL routing for mobile low power wireless sensor networks using corona mechanism, in: Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on, 2014, pp. 200–209, doi:10.1109/SIES.2014.6871205.
[15] O. Gaddour, A. Koubâa, {RPL} in a nutshell: a survey, Comput. Netw. 56 (14) (2012) 3163–3178, doi:10.1016/j.comnet.2012.06.016.
[16] C. Gomez, A. Boix, J. Paradells, Impact of LQI-based routing metrics on the performance of a one-to-one routing protocol for ieee 802.15.4 multihop networks, EURASIP J. Wirel. Commun. Netw. 2010 (2010) 6:1–6:20, doi:10.1155/2010/205407.
[17] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, IEEE, 2000.
[18] K.-S. Hong, L. Choi, Dag-based multipath routing for mobile sensor networks, in: ICT Convergence (ICTC), 2011 International Conference on, 2011, pp. 261–266, doi:10.1109/ICTC.2011.6082593.
[19] K. Kim, G. Montenegro, S. Park, I. Chakeres, C. Perkins, Dynamic MANET on-demand for 6LoWPAN (dymo-low) routing, Internet Eng. Task Force, Internet–Draft, 2007a, pp. 1–17.
[20] K. Kim, S.D. Park, G. Montenegro, S. Yoo, N. Kushalnagar, 6LoWPAN ad hoc on-demand distance vector routing (load), Netw. WG Internet Draft 19 (2007b). (work in progress).
[21] K. Kim, S. Yoo, J. Park, S.D. Park, J. Lee, Hierarchical routing over 6LoWPAN (hilow), draft-daniel-6LoWPAN-hilow-hierarchical-routing-01, 2007, pp. 1–12.
[22] J. Ko, A. Terzis, S. Dawson-Haggerty, D. Culler, J. Hui, P. Levis, Connecting low-power and lossy networks to the internet, Commun. Mag., IEEE 49 (4) (2011) 96–101, doi:10.1109/MCOM.2011.5741163.
[23] I. Korbi, M. Ben Brahim, C. Adjih, L. Saidane, Mobility enhanced RPL for wireless sensor networks, in: Network of the Future (NOF), 2012 Third International Conference on the, 2012, pp. 1–8, doi:10.1109/NOF.2012.6463993.
[24] N. Kushalnagar, G. Montenegro, C. Schumacher, IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals, 2007, (RFC 4919 (Informational)).
[25] K. Lee, R. Sudhaakar, L. Dai, S. Addepalli, M. Gerla, RPL under mobility, in: Consumer Communications and Networking Conference (CCNC), 2012 IEEE, 2012, pp. 300–304, doi:10.1109/CCNC.2012.6181106.
[26] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The Trickle algorithm, 2011, (RFC 6206 (Proposed Standard)).
[27] M. Maleki, K. Dantu, M. Pedram, Power-aware source routing protocol for mobile ad hoc networks, in: Proceedings of the 2002 International Symposium on Low Power Electronics and Design, in: ISLPED '02, ACM, New York, NY, USA, 2002, pp. 72–75, doi:10.1145/566408.566429.
[28] J. Martocci, P.D. Mil, N. Riou, W. Vermeylen, Building automation routing requirements in low-power and lossy networks, 2010, (RFC 5867 (Informational)).
[29] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, 2003, (RFC 3561 (Experimental)).
[30] K. Pister, P. Thubert, S. Dwars, T. Phinney, Industrial routing requirements in low-power and lossy networks, 2009, (RFC 5673 (Informational)).
[31] R.C. Shah, J.M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in: Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, 1, IEEE, 2002, pp. 350–355.
[32] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, K. Leung, A survey on the IEFT protocol suite for the internet of things: standards, challenges, and opportunities, Wirel. Commun., IEEE 20 (6) (2013) 91–98, doi:10.1109/MWC.2013.6704479.
[33] J.P. Singh, N. Bambos, B. Srinivasan, D. Clawin, Wireless LAN performance under varied stress conditions in vehicular traffic scenarios, in: Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th, 2, IEEE, 2002, pp. 743–747.
[34] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, 1998, pp. 181–190.
[35] J. Tripathi, J. de Oliveira, J. Vasseur, A performance evaluation study of RPL: routing protocol for low power and lossy networks, in: Information Sciences and Systems (CISS), 2010 44th Annual Conference on, 2010, pp. 1–6, doi:10.1109/CISS.2010.5464820.
[36] N. Tsiftes, J. Eriksson, A. Dunkels, Low-power wireless IPv6 routing with contikirpl, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, in: IPSN '10, ACM, New York, NY, USA, 2010, pp. 406–407, doi:10.1145/1791212.1791277.
[37] T. Watteyne, A. Molinaro, M.G. Richichi, M. Dohler, From MANET to IETF roll standardization: a paradigm shift in wsn routing protocols, Commun. Surv. Tut., IEEE 13 (4) (2011) 688–707.
[38] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 routing protocol for low-power and lossy networks, 2012, (RFC 6550).
[39] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, A performance analysis of point-to-point routing along a directed acyclic graph in low power and lossy networks, in: Network-Based Information Systems (NBiS), 2010 13th International Conference on, 2010, pp. 111–116, doi:10.1109/NBiS.2010.65.
[40] M. Younis, M. Youssef, K. Arisha, Energy-aware routing in cluster-based sensor networks, in: Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on, IEEE, 2002, pp. 129–136.



Afonso Oliveira is a PhD student at Instituto Superior Técnico (IST), from University of Lisbon, and is also a Junior Research Fellow in the Communication Networks and Mobility group at INESC-ID, an Associate Laboratory of Portugal. He received his MSc from IST in computer networks with a thesis on design and optimization of optical networks. His current research interests include networks running on low-power such as low-power and lossy networks (LLNs) and body area networks (BANs).

Teresa Vazão is an associate professor at the Department of Electrical and Computer Engineering, at Instituto Superior Técnico (IST), from University of Lisbon, and is also a senior researcher at INESC-ID, an Associate Laboratory of Portugal. She received her PhD from IST in electrical and computer engineering, with a specialization in computer networks. She has developed research in the field of communications networks and her currently research interests are focused in sensor and vehicular networks, with a special emphasis in routing and performance aspects. She had supervised more than 40 students. Her research work, resulted in international publications in different forums, international journals, conferences and patents. She coordinated INESC-ID participation in several European projects, like MIAVITA – Wireless Sensor Network for Volcanic Monitoring and SENSATION - Advanced Sensor Development for Attention, Stress, Vigilance and Sleep/Wakefulness Monitoring. She also participated, as team member, in several international and national projects.