RPL Routing Protocol in Advanced Metering Infrastructures: an Analysis of the Unreliability Problems

Emilio Ancillotti, Raffaele Bruno, Marco Conti
Institute for Informatics and Telematics (IIT)
Italian National Research Council (CNR)
Via G. Moruzzi 1, 56124 Pisa, ITALY
Email: {a.ancillotti,r.bruno,m.conti}*@iit.cnr.it*

***Abstract*—To allow pervasive and distributed monitoring and control of grid devices and resources, the next-generation electricity grid needs a scalable and reliable two-way communication infrastructure, known as Advanced Metering Infrastructure (AMI). In such communication system, characterized by the interconnection of thousands of resource-constrained embedded devices, such as smart meters and intelligent electric devices (IEDs), the routing protocol plays an essential role to guarantee a timely and reliable communication service. In this paper we investigate the performance of RPL, the IPv6 routing protocol recently standardized by IETF to meet the requirements of such networks. Special emphasis is given to the analysis of route-level attributes, such as path stretch, route lifetimes, dominance, and flapping, which are used to estimate the quality and stability of the RPL routes. Our results show that RPL nodes may suffer from severe unreliability problems, mainly because RPL often selects sub-optimal paths with low quality links. Furthermore, RPL routes are strongly dominated by a single route, and this may prevent RPL from quickly adapt to link quality variations. We believe that the findings of this study may facilitate the design of new mechanisms to improve the reliability and adaptability of RPL.**

***Keywords*-AMI, RPL, reliability, stability, Contiki OS, performance evaluation.**

I. Introduction

It is foreseen that the next-generation electricity grid, known as "*smart grid*", will be more resilient, dynamic, and adaptive, as well as economically and environmentally sustainable, than the current power systems [11]. These benefits will be achieved through the modernization of the electrical infrastructure, the massive adoption of distributed generation technologies based on renewable energy resources [2], and the integration of widely dispersed energy storage systems (i.e., batteries) to complement renewable energy generation [37]. These technological innovations entail a considerable transformation of the way electricity production and distribution will be administered and optimized in the future grid, evolving from a centralized to a fully distributed approach [27]. Furthermore, new management and control mechanisms are needed, especially at the distribution system level, to dynamically adapt the electricity usage of end customers to variable energy supply conditions. This can be achieved with *demand response* (DR) techniques, which allow customers to adjust their electricity usage patterns to changes in the price of electricity over time, or *direct load control* (DLC) techniques, which allow utilities to schedule power consumption of residential appliances [28], [36].

To enable pervasive, wide-area monitoring and distributed control of a massive number of grid resources, there is widespread consensus that the deployment of a *scalable and reliable two-way communications and networking infrastructure* is essential to the smart grid. This communication infrastructure, tightly integrated with the electric infrastructure, is needed to enable the collection of real-time and fine-grained measurements about the status of the electric devices, the dispatch of configuration instructions, and the support of coordinated and distributed control functions [33]. This network, commonly denoted as *Advanced Metering Infrastructure* (AMI) [16], will primarily consist of smart electric meters deployed at individual households, which will be used as *sensors* to monitor electric grid quality and power consumption of home appliances, as well as *home energy management systems* (HEMSs) to support applications such as demand response and electric vehicle recharging [13]. Typically, these smart meters will be interconnected using heterogeneous technologies, ranging from power-line communications [12] to a combination of wireless technologies (e.g., IEEE 802.15.4 or low power WiFi [15]). In addition, in a typical AMI deployment, groups of meters within physical proximity (e.g., apartment buildings in urban centers) will be interconnected using multi-hop mesh networks that, in turn, will be connected to a backhaul network through one or more gateways. Then, the AMI system can be used to transmit metering data towards network aggregation points, or to send commands issued by utility-based controllers to electric appliances [33].

It is straightforward to observe that the routing protocol plays a crucial role in AMI networks to guarantee a timely and reliable communication service. However,

978-3-901882-46-3 ©2012 IFIP

it can be difficult to design a routing protocol able to meet the communication requirements of AMI networks [7]. First of all, typical smart meters are resource-constrained embedded devices with limited processing power and storage capabilities. Furthermore, in AMI networks the links between the devices are generally characterized by high loss rates, low data rates, and instability due to the unplanned network deployments and the use of low-power link-layer technologies. From this perspective, an AMI system is an instance of the more general class of the so-called *Low power and Lossy Networks* (LLNs). Among the many efforts for designing suitable routing protocols for LLNs, the most consolidated and widely recognized proposal is RPL [34], a new IPv6 routing protocol standardized in [35] by the IETF ROLL working group, which is intended to meet the requirements of a wide range of LLN application domains, including building automation [21], urban sensor networks [8], and large-scale AMI systems [25]. RPL is a gradient-based routing scheme that organizes the network topology as a Directed Acyclic Graph (DAG)[1], where each node selects the next hop as the neighbor (called *preferred parent*) that offers the largest gradient to the destination acceding to some metric(s) (more details in Section III). This routing structure is optimized for traffic to or from one or more root nodes (or sinks in WSN jargon). Given the importance of RPL applications, recently a few papers have addressed the performance evaluation of RPL in different use cases [1], [4], [17], [30], focusing on protocols overheads, network set-up latencies, network throughput, and end-to-end packet delays. However, the ability of RPL to meet the stringent reliability requirements of AMI systems has not been sufficiently investigated.

The purpose of this paper is to explore the performance of RPL in a typical AMI system used for collecting metering data. More specifically, we aim at understating the behavior of RPL and identifying the *major causes of unreliability* that may be experienced by a non-negligible fraction of nodes in the network. A special emphasis will be dedicated to investigate the *routing stability* and *path quality* in terms of route-level attributes, such as path stretch, route lifetimes, route dominance, and the characteristics of route flapping. Our results show that RPL nodes may suffer from severe unreliability problems mainly because RPL measures the quality of a link only when data traffic is sent through that link. This can frequently lead to inefficient parent selection and, consequently, to the use of sub-optimal paths including low quality links. Furthermore, RPL routes are strongly dominated by a single route, and this may prevent RPL from quickly adapt to link quality variations. Finally, route changes do not provide any improvement of the network path quality, but they are mostly due to temporary routing inconsistencies. In other words, not only RPL fails in selecting reliable network paths, but it also unable to quickly recover from wrong routing decisions. It is important to observe that routing stability and its impact on network performance has been studied previously for other routing protocols designed for wireless multi-hop environments [26]. To the best of our knowledge, this paper presents the first characterization of routing stability in RPL-based networks. We also believe that the findings of this study may facilitate the design of new mechanisms to improve the reliability and adaptability of RPL.

The rest of this paper is organized as follows. Section II describes the findings of recent performance studies on RPL. In Section III we present an overview of the main features of the RPL standard. Section IV-A describes the implementation of RPL in Contiki OS. Section IV reports and explains the simulation results used to assess the applicability of RPL in typical AMI deployments. Finally, Section V draws conclusions and discusses future work.

II. Related Work

Recently a number of papers have been published focusing on the performance evaluation of RPL in different uses cases. In [1], the Contiki COOJA simulator is used to analyze the signaling overheads and the network setup delays. The shown results indicate that network setup times are generally limited but this is achieved at the expense of high protocol overheads, which can be a significant portion of the overall network traffic. On the other hand, an experimental campaign has been carried out in [4] using a sensor network featuring 80 nodes, which was designed to collect environmental experimental data. Results show that with a proper parameter tuning, RPL can quickly discover the network topology and build network routes. A simulation study is conducted in [30] to evaluate the quality of RPL routes (e.g., in terms of path stretch with respect to shortest paths) with point-to-point traffic. Results indicate that in the considered scenarios the path quality of RPL is not drastically worse than the optimal one. In addition, in [30] it is also shown that local repair mechanism are much quicker than global repair mechanisms in fixing connection outages. The performance of the RPL routing protocol with bi-directional traffic flows is also investigated in [17]. Results indicate that a different routing protocol, called LOAD, which is a lightweight variant of AODV, can provide similar data delivery ratios as RPL but inducing less protocol overheads. Finally, authors in [6] point out potential weaknesses of RPL such as: *i*) the possibility of loops, *ii*) the implicit assumption on the bi-directionality of links, and *iii*) packet fragmentation issues due to the use of constrained link layers, such as IEEE 802.15.4, with small MTUs, which

[1]A DAG is a directed graph wherein all edges are oriented in such a way that no cycles exist.

calls for further developments of the RPL specifications. However, to the best of our knowledge there exist no comprehensive studies of RPL routing reliability and stability in AMI networks, which is the focus of this paper.

Other papers have proposed enhancements for the RPL draft standard to improve routing performance, or to support additional features. For instance, authors in [5] explore optimized root-initiated broadcast mechanisms for RPL-based networks, which take advantage of the tree-like network topology maintained by RPL. In [24], the authors propose extensions to the IEEE 802.15.4 MAC protocol to allow nodes to associate to multiple parents, and they take advantage of this modified MAC layer to extend RPL with the possibility of forwarding packets over multiple paths. In this way, traffic is spread more uniformly over all possible parents, leading to a slight increase of packet delivery ratios. In [32] a number of modifications to RPL are proposed, such as a new ETX-based rank computation method and a reverse path recording mechanism, to make RPL more robust against fading. Finally, connectivity detection and channel scanning procedures are proposed in [19] to support self-configuration and utilization of multiple channels in RPL-based wireless mesh networks.

We believe that a better understanding of routing stability and reliability will be beneficial to design modifications to the RPL routing engine to achieve a better tradeoff between routing adaptability and routing efficiency.

III. RPL Overview

The design principles of the RPL routing protocol are driven by the primary goals of: *i*) minimizing the memory requirements (i.e. routing table sizes) and the complexity of the routing functionalities to facilitate the deployment on micro-controllers with constrained resources; *ii*) reducing the signaling overheads to lower both the bandwidth usage and the energy consumption due to routing procedures; and *iii*) distribute compact routing information to support link-layer technologies with restricted frame sizes. Furthermore, in the RPL design it is assumed that most traffic is *multipoint-to-point*, i.e. it flows from the RPL nodes (the smart meters in AMI systems) to a few controllers (the sinks in WSN jargon), while it is less frequent to have traffic from the controllers to the RPL nodes or between the RPL nodes.

As observed in Section I, to minimize the network state information, RPL organizes the network topology into multiple, non-overlapping "Destination-oriented" Directed Acyclic Graphs (DODAGs), one for each sink, or DODAG root, as shown in Fig. 1. Then, each node adopts an objective function to compute a scalar value, called *rank*, which represents the abstract distance of that node from the DODAG root according to some metric(s). In other words, the objective function is used

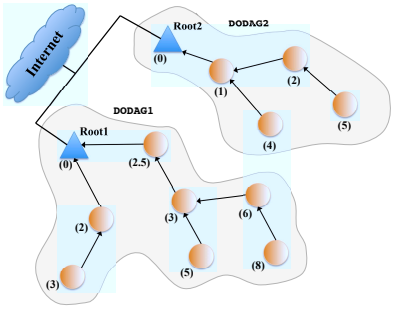

Figure 1. Example of an RPL network with multiple DODAGs. Values in parenthesis are nodes' rank. Solid arrows indicate child-parent relationship.

to define how RPL nodes select and optimize routes to meet the application QoS requirements [31]. Nevertheless, the node's rank must satisfy generic properties independently of the objective function. More precisely, the rank of nodes along any path to the DODAG root should be incremented in a strictly monotonic fashion to avoid routing loops. In addition, RPL restricts the ability of a node to change its membership in the DODAG to limit routing instability. For instance, an RPL node cannot assume a greater rank (i.e. move deeper in the DODAG) than previously obtained. In general, the stability of the rank influences the stability of the routing topology. We will elaborate more in depth on this observation in the following sections.

A. DODAG construction and upward routes

To build a DODAG, a root node advertises its presence by periodically sending control messages, called DODAG Information Object (DIO), as options of IPv6 Router Advertisements, which are transmitted using multicast frames. In addition, the DIO messages list the common configuration attributes that all RPL nodes should adopt in the DODAG. Upon receiving DIO messages, each RPL node learns the set of its neighbors and their rank values.

The objective function associated to the DODAG specifies: *a*) how RPL nodes receiving the DIO messages should compute their rank from the one of their own neighbors, and *b*) how RPL nodes should select a *preferred parent* node among their neighbors having a lower rank, which will serve as default next hop towards the DODAG root. To disseminate routing information, the RPL nodes transmit new DIO messages using the flooding algorithm described in Section III-C. Alternatively, RPL nodes may multicast DODAG Information Solicitation (DIS) messages to probe their neighborhood and solicit DIO messages. Finally, the root can trigger the recalculation of the entire DODAG by simply increasing a sequence number, the DODAG version, in the emitted DIO messages (e.g. to initiate a global repair

procedure). RPL also supports mechanisms, based on DIO messages, which may be used for local repair within the same DODAG version (e.g., if a loop is detected).

It is important to note that DIO messages in RPL have not the same purpose of HELLO messages in other MANET routing protocols. First of all, RPL nodes do not verify that they can communicate with the neighbors from which they have received a DIO message. Secondly, *DIO messages are not used to monitor links*. Therefore, RPL must rely on external mechanisms in order to verify link properties and parent reachability. The IPv6 Neighbor Unreachability Detection (NUD) mechanism [22] is one of the options considered in the RPL standard.

B. Downward routes and destination advertisements

A DODAG can be used only to support upward traffic, because the RPL nodes store in their routing tables only the next-hop node to forward traffic destined to the root. In order to also support downward traffic RPL nodes must generate Destination Advertisement Object (DAO) messages to propagate destination information upward along the DODAG. More specifically, RPL assumes that one or more *target* devices may be associated with (owned by) an RPL node (e.g., home appliances associated with the smart meter running the RPL daemon), and DAO messages are periodically generated by that RPL node to advertise the addresses of its target devices. The way DAO messages are transmitted and processed by intermediate routers depends on the mode of operation. In the first mode of operation, called *non-storing mode*, an RPL node with associated targets sends unicast DAO messages to the DODAG root. In this case, the DAO message also includes information on the parent set of the RPL node that generated the message. Then, upon receiving DAO messages from all the RPL nodes along a path, the DODAG root can construct a source route to the advertised destinations by reclusively looking at the DAO parent information. In the second mode of operation, called *storing mode*, each RPL node sends unicast DAO messages to a subset of its parents, which store in their routing tables the address of that RPL node as the next-hop node to reach the advertised targets (i.e. they record the reverse path information from the DAO message). On receiving a unicast DAO, a node must generate a new DAO and transmit it to its parents to ensure that routing information propagates in the network. Then, classical hop-by-hop routing is used by RPL nodes for reaching destinations learned from DAOs.

It is intuitive to observe that the downsides of the non-storing mode are that: *i*) the DODAG root has to maintain in its routing table as many source routes as destinations in the network, *ii*) the additional source routing headers increases the probability of fragmentation of also the data packets [6], and *iii*) if one RPL node is temporarily unreachable, its entire sub-DODAG – the set of other nodes whose paths to the DODAG root pass through that node – will be unreachable. On the other hand, in storing mode each RPL node must store routing information to reach all the destinations that are in its own sub-DODAG, which may be too demanding for the limited memory resources of small embedded devices.

C. RPL use of the Trickle algorithm

The Trickle algorithm, specified in [20] is used to control the sending rate of RPL messages. Basically, the Trickle algorithm implements a consistency check model to verify if RPL nodes have out-of-date routing information. If not, the Trickle algorithm will exponentially reduce the rate at which DIO messages are emitted. Otherwise it quickly schedules the transmission of a DIO message to update this information. More precisely, an RPL node schedules the transmission of a DIO message at a given time t uniformly sampled within the Trickle timer interval. During the period t, the RPL node counts all DIO messages that convey routing information consistent with its own network state. If the number of consistent DIO messages received by time t is below a pre-configured redundancy threshold, called $DIO\_REDUNDANCY\_CONSTANT$, the scheduled DIO message is transmitted. Otherwise, the DIO transmission is cancelled and the Trickle timer is doubled (up to a maximum value $DIO\_INTERVAL\_MAX$), and a new DIO transmission is scheduled. On the other hand, RPL resets the Trickle timer to a minimum value $DIO\_INTERVAL\_MIN$ if there is an inconsistent condition, such as the reception of a DIO message with out-of-date information.

IV. Performance Evaluation

The operations of an RPL-based AMI network are simulated using the Contiki COOJA simulator [23]. In the following we first present the main characteristics of the RPL implementation in Contiki. Then, we describe the simulation settings and we show the simulation results.

A. Contiki Implementation of RPL

Contiki is an open source operating system specifically designed for embedded systems (e.g. sensors) with small amounts of memory to be used mostly in applications for the Internet of Things domain [10]. Among the many protocols and mechanisms supported in Contiki, relevant to our study are the 6LowWPAN header compression [18] and the IETF RPL routing [35] (hereafter ContikiRPL[2]). In addition, Contiki includes COOJA, a *cross-level* simulation platform that combines low-level simulation of device hardware with the simulation of high-level protocols [23]. In other words, COOJA allows simulating compiled Contiki software readily deployable in actual hardware.

[2]http://www.contiki-os.org

ContikiRPL includes all the fundamental mechanisms specified in the standard [35]. However, it does not support the non-storing mode (i.e., source routing) and the security features. Furthermore, the RPL standard has left several mechanisms underspecified, which are considered implementation dependent. Thus, the goal of this section is to clarify how these mechanisms are implemented in ContikiRPL, since this will affect the behaviors observed in the simulations. First of all, ContikiRPL relies on the IPv6 Neighbor Discovery (ND) protocol [22] for address resolution (i.e., ARP-like functionalities) and for periodically verifying that a neighbor is still reachable via a cached link-layer address (i.e., ICMP-like functionalities). More precisely, when a timer (set to 600 seconds by default) expires ContikiRPL checks if a neighbor is still reachable by sending one or more solicitation messages. It is also important to note that the neighbor cache used by the Neighbor Unreachability Detection (NUD) mechanism [22] has a limited size (20 entries by default) and that, when the neighbor cache is full, the oldest entry is removed if a new neighbor is detected. It is intuitive to observe that during the DODAG construction process the DODAG root is generally the first node to be added to the neighbor cache of other RPL nodes that are close to it. Thus, in our experiments it occurred that the root was temporary removed from the neighbor cache of some RPL nodes. To avoid this inconsistent behavior we have enhanced the NUD implementation used by ContikiRPL to replace the least-recently-used entry in the neighbor cache instead of the oldest one.

Regarding the objective function used to compute nodes' rank values, ContikiRPL adopts a variant of the *Objective Function Zero* (OF0) specified in RFC 6552 [29]. OF0 is the default objective function of RPL and its purpose is to select a preferred and backup parent in such a way to find the nearest DODAG root. More specifically, OF0 determines the rank $R(N)$ of an RPL node N by adding a strictly positive, normalized scalar value to the rank $R(P)$ of its selected preferment parent P using the following formula:

$$R(N) = R(P) + (Rf * Sp + Sr) * \text{MinHopRankIncrease}\ , \quad (1)$$

where Sp describes the dynamic properties (e.g. in terms of ETX value [9]) of the link between N and P, while Sr and Rf are normalization parameters (in ContikiRPL we have $Rr = 1$ and $Sr = 0$). The MinHopRankIncrease factor guarantees that the rank difference between a node and its parent is greater than the minimum defined in the RPL standard [35]. It is important to note that in ContikiRPL, the preferred parent of an RPL node is not the neighbor with the smaller rank, but it is the neighbor with the shortest path in terms of ETX towards the DODAG root. In addition, ContikiRPL adopts a simple hysteresis mechanism to reduce parent switches in response to small metric changes[3]. This is in line with what recently proposed in [14].

Finally, concerning the ETX computation, ContikiRPL initializes to five (i.e., the maximum number of link-layer retransmissions) the ETX value of newly discovered links, while the MAC driver updates the ETX value *only* for the links on which data traffic is transmitted. Then, ContikiRPL maintains a small (12 entries by default) neighbor attribute cache, where each entry is a moving average of the ETX samples produced by the MAC driver. It is important to observe that the MAC driver does not pass to the routing protocol the ETX values for neighbors that are not listed in this cache. We have also enhanced ContikiRPL to manage the neighbor attribute cache according to a least-recently-used replacement policy.

Table I
RPL PARAMETERS.

Parameter	Value
DIO_INTERVAL_MIN	8 ms
DIO_INTERVAL_MAX	2.3 hours
DIO_REDUNDANCY_CONSTANT	10

B. Simulation settings

In this study we have simulated a network with a single sink node, and we have assumed that the smart meters are randomly scattered in a squared area with a side $L =$ 800 meters and with the sink placed at the center. The main default RPL parameters used in the simulations are listed in Tab. I.

The number of smart meters in the simulated area is varied within the set $[50, \ldots, 150]$, which allows us to analyze network scenarios from low to high density. To take into account the resource (e.g., memory) limitations of real hardware, we have used COOJA to *emulate* a Tmote Sky platform, an MSP430-based board with an 802.15.4-compatible CC2420 radio chip. Furthermore, to simulate realistic interference we have used the Multi-path Ray-tracer Medium (MRM) model supported by COOJA, which utilizes ray-tracing techniques to model various radio propagation effects (e.g., multi-path, refraction, diffraction, etc.) [3]. The transmitted power of each RPL node is set to the minimum value that ensures successful transmissions within a distance of 150 m if there is no interference or channel noise. As far as the MAC protocol is concerned, we use a classic CSMA/CA random scheme shipped with the Contiki kernel. It is important to note that investigating RPL energy consumption is out of the scope of this paper, because it is generally difficult to accurately quantified it in network simulations. In addition, it is also reasonable

[3]In ContikiRPL, a new parent is selected only if its minimum-cost path to the root is shorter (in terms of ETX) than the current path by at least 0.5.

to assume that smart meters have less energy concerns than other sensors used for environmental monitoring. Thus, no power saving scheme is operating on the RPL nodes in the simulations. A low-intensity metering traffic is simulated by letting each RPL node to generate a constant-bit-rate (CBR) traffic of 30-byte long messages per minute. It is useful to point out that data rates depend on the application requirements, but typical DR and DLC applications are foreseen to require one metering reading every few minutes, while control applications used to preserve electricity quality, such as frequency regulation, will require to monitor grid resources on a higher frequency [7].

All results presented in the graphs are averaged over 10 simulation runs, and error bars show the 95% confidence level. Each experiment simulates four hours of network operations, and statistics are collected after removing the transient phase (i.e. one hour of simulation time).

C. Evaluation methodology and performance metrics

Our analysis of the routing and network performance is based on the parent set information collected from each RPL node in the network and the link quality information. More specifically, the average link delivery ratios between all pairs of nodes are determined by the radio propagation model and the parameters of the radios and they can be computed at the network initialization time. These delivery ratios are used to compute the ETX cost, which is given by the equation: $1/(d_f * d_r)$, where d_f and d_r are the links delivery ratios in the forward and reverse directions, respectively [9]. From the knowledge of the *full network graph* and the associated link costs we compute the shortest paths between the RPL nodes and the DODAG root using an implementation of the Dijkstra's shortest-path algorithm. On the other hand, every time a preferred parent is selected or changed during the simulation by any RPL node we trace the time of this event, and the addresses of both the child node and the parent node. This information univocally identifies the links used in the RPL network at any point in time. From the knowledge of the *RPL network topology* we compute the actual upward routes used by RPL nodes to reach the DODAF root, and the cost of these paths. For each one of the observed RPL upward routes we also keep track of how many times its has been used, and each time it has been used how long it lasted. It is also important to notice that a single preferred parent change can cause the modification of many RPL upward routes.

Let us define the *dominant route* for an RPL node as the RPL upward route that has been used most of the time during the simulation. Then, similarly to the work in [26] we use the following parameters to analyze the routing performance in terms of routing stability and path quality:

- *prevalence of the dominant route*, defined as the fraction of total simulation time during which the dominant route is used;
- *persistence of the dominant route*, defined as the average duration of the dominant route before a route change;
- *route flap*, which refers to a route change. In the following section we describe in mode details the metric used to quantify the effect of a route flap on the ETX cost of the new RPL upward route;
- *path stretch*, defined as the ratio between the cost (measured in terms of hop or total ETX) of the RPL upward route used by each RPL node and the shortest path (in terms of ETX cost).

On the other hand, to analyze the network performance we use the following classical metrics:

- *end-to-end delay*, defined as the average time to successfully transmit a packet from an RPL node to the DODAG root;
- *packet delivery ratio*, defined as the fraction of packets sent by an RPL node that is lost in the network;

It is important to point out that the packet delivery ratios are a fundamental metric to validate if RPL is suitability for AMI systems. However, not only average loss rates are important but also distributions are relevant. Indeed, even if the average loss rates are within the bounds demanded by the metering application it is necessary to check that there are not individual network devices that experience high loss rates, because this would make impossible to ensure reliable management of the grid resources connected to that smart meter.

D. Numerical results

First of all, let us consider the path quality and stability of the RPL upward routes. As a matter of fact, the DODAG construction mechanisms cannot ensure that shortest paths, in terms of total ETX, are always selected. Fig. 2 shows the average path stretch values for the RPL routes used by RPL nodes that are at the same minimum distance (in terms of hops) from the DODAG root. The results indicate that the average path stretch is fairly limited, with RPL using network paths that are, *on average*, less than 20% worse that the optimal ones. Furthermore, the path stretch values for different distances from the DODAG root are fairly close, which suggests that RPL nodes that are far from the DODAG root not necessarily experience worse path qualities than nodes that are closer to the DODAG root. It is also useful to point out that these results are in line with the ones presented in [17], [30]. However, simply looking at the average path stretch might result in overlooking critical localized behaviors, and in failing to properly evaluate routing inefficiencies that might be severe, although temporary. For these reasons, in Fig. 3

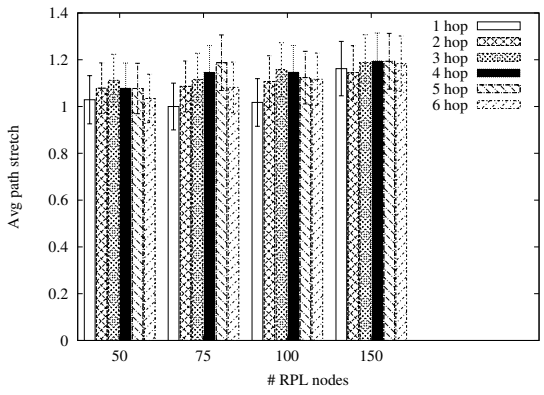

Figure 2. Average path stretch (in terms of ETX) versus the hop distance between RPL nodes and DODAG root.

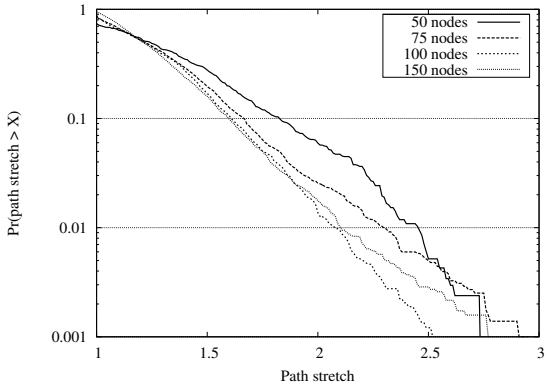

Figure 3. CCDF of path stretch (in terms of ETX).

we report the complementary cumulative distribution function (CCDF) of the path stretch values for all the RPL routes that are used during the simulations. Surprisingly, the plots indicate that there is a non-negligible probability that RPL nodes use upward paths that are significantly longer than the shortest ones. More specifically, in the considered scenarios more than 10% of the RPL nodes may use RPL routes with a path stretch that ranges from 60% (in 150-node topologies) to 85% (in 50-node topologies). To summarize, while the average "quality" of RPL routes is not significantly worse than the optimal case, there is a non-negligible probability that some RPL nodes can use longer network paths. The main reason is that *RPL has a very partial knowledge of the link qualities because the neighbor attribute cache has a limited size, and RPL collects ETX estimates only for the links on which data traffic is transmitted*. Later in this section we will analyze how this condition affects the reliability of network communications.

Let us now focus on characterizing the stability of the RPL routes. Our aim is to verify if the observed trends in path stretch metric are the result of a clear and stable choice of a sub-optimal route between the RPL nodes and the DODAG root, or, instead, they are due to frequent route changes (route flap). To this end, Fig. 4 and Fig. 5 show the CCDFs of the prevalence and of the persistence, respectively, of the dominant RPL upward route for different network sizes. From the shown results we can draw two main observations. The first one is that the *dominant RPL routes are remarkably prevalent* over all the other routes used by each RPL node[4]. More specifically, Fig. 4 shows that the median prevalence (i.e., the prevalence value experienced by the dominate routes of at least 50% of the RPL nodes) range from 84% (in 150-node topologies) to 96% (in 50-node topologies). The second important observation is that *dominant RPL routes are long-lived*. More specifically, the dominant RPL routes have a median persistence that ranges from 1.67 hours (in 76-node topologies) to 2.2 hours (in 50-node topologies). In addition, more than 15% of the RPL nodes in the network have an average duration of the dominant RPL routes that is at least 3.8 hours, which is 98% of the total simulation time. To explain these results on route persistence it is necessary to recall that, although RPL allows attachment to a "better parent" at any time, the hysteresis mechanism described in Section IV-A avoids that parent switches occur for small metric changes.

Although dominant RPL routes are long-lived, route changes may occur. When there is a route flap, the new discovered route may lead to either an improvement in the path quality compared to the route used before the route change, or to a degradation of the path quality. In order to quantify the path quality variation after a route flap we record the links with the greatest ETX value for each RPL route used by a node, which can be considered as the *worst link* in the path. Then, we calculate the percentage variation of link cost for the worst link after a route flap. Fig. 6 shows the CCDF of this metric for varying networks sizes. The results indicate that about 15% of route flaps offer an improvement of the worst link, another 30% of route flaps induce negligible variation of the works link, while the remaining route flaps cause a significant degradation of the path quality. It is also useful to observe that the RPL routes the include links with very low quality (i.e., high ETX values) are generally shorter, in terms of hops, than the optimal paths that minimize the total ETX cost. The reason is that this low quality links are generally shortcuts in the network. This also explains why the average degradation of path stretch with RPL is not significant.

To conclude our analysis, in the following we evaluate the impact of RPL inefficiencies on the average packet delivery ratios per node and the average packet end-to-end delays, which are shown in Fig. 7 and Fig. 8, respectively. The results indicate that a few RPL nodes are not able to reliably transmit their data to the DODAG root. More specifically, the 95% percentile of packet

[4]Note that this prevalence behavior is radically different from the one observed in [26] for OLSR.

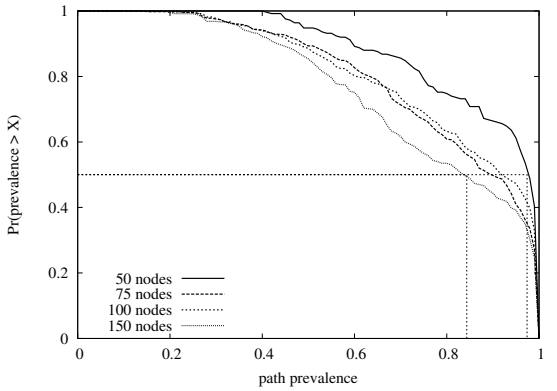

Figure 4. CCDF of path prevalence for the dominant RPL upward route.

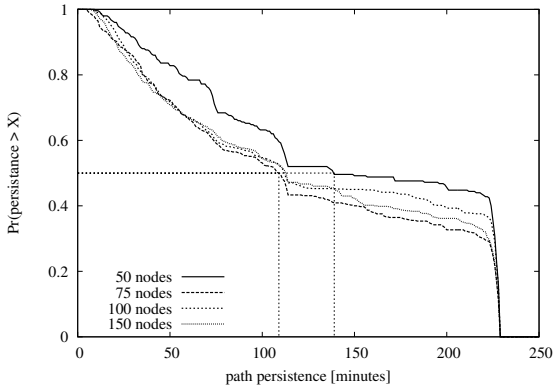

Figure 5. CCDF of path persistence for the dominant RPL upward route.

delivery ratios ranges from 0.05 (in 50-node topologies) to 0.1 (in 150-node topologies). Nevertheless, a few nodes can experience up to 0.3 packet delivery ratios It is important to observe that such packet delivery ratios are not meeting the communication requirements of AMI systems [7] because, every smart meter should be able to reliably deliver its metering data to the network data aggregator to guarantee a correct management and control of the smart grid services. On the other hand,

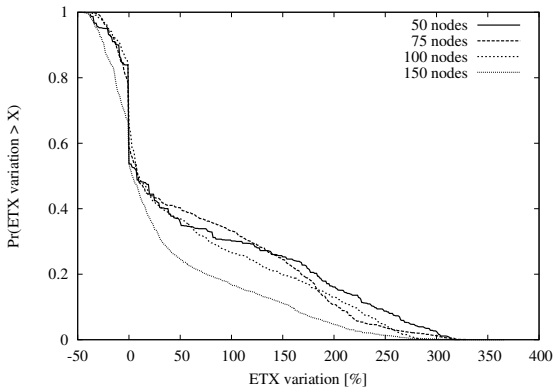

Figure 6. CCDF of ETX variation for the worst link in the path after a route flap.

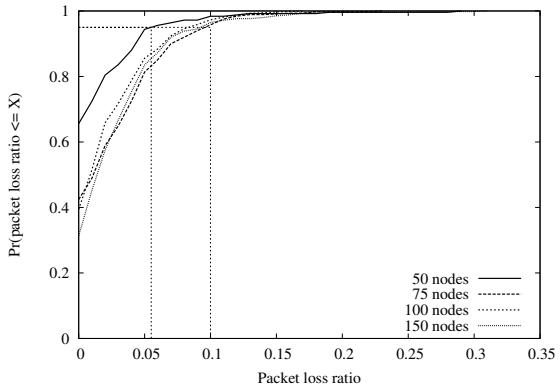

Figure 7. CDF of packet loss ratios.

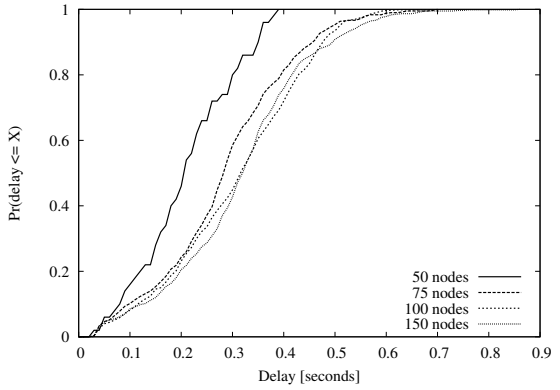

Figure 8. CDF of packet delays.

as shown in Fig. 8 packet end-to-end delays are very limited and within one second. These latencies are not negatively affecting typical DR and DLC applications, which operate on a time scale of the order of minutes.

We have investigated in depth the simulation traces to identify the primary causes of these excessive packet loss rates. Our results (not shown here due to space limitations) indicate that that the number of packet losses due to buffer overflows or channel contention are

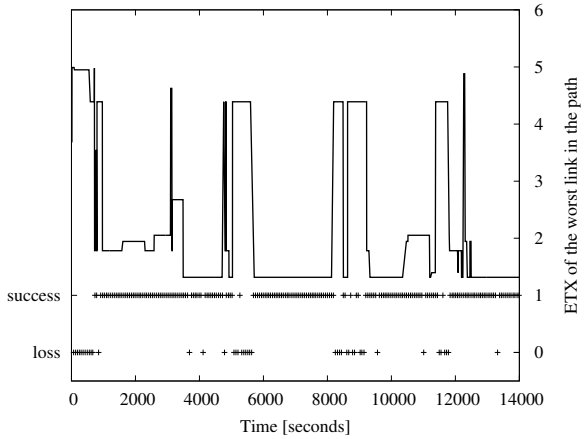

Figure 9. Example of temporal distribution of packet losses and ETX of the worst link in the path for node 2.

negligible. On the contrary, most of the packet losses occur on low-quality links that may be selected by RPL even if alternative high-quality links are available. To clarify this concept, we analyze the packet losses of node 2, which experiences a packet loss ratio close to 0.2. More specifically, in Fig. 9 we show the temporal distribution of packet losses for node 2 during one simulation run. In the same figure we also plot the ETX of the worst link in the RPL route used by node 2 to deliver its traffic to the DODAG root. Interestingly we can observe that the packet losses are not uniformly distributed during the simulation period, as it would be expected in case such losses were due to collisions. On the contrary, packet losses are mainly concentrated in the time intervals when RPL select upward routes that include a low-quality link affected by a high loss rate, i.e. high ETX. It is also important to note that in "normal" conditions the worst link of the RPL route used by node 2 has an ETX value lower than two. Thus, selecting an RPL route including a worst link with an ETX value greater than four represents a significant degradation of the path overall reliability. To conclude, our results indicate that packet losses are mostly due inefficient routing decisions, and that system reliability could be improved by avoiding the unnecessary use of low-quality links.

V. Conclusion

In this paper we have investigated the performance of the RPL routing protocol in AMI networks, focusing on routing satiability and reliability performance. Our results clearly indicate that the RPL route selection procedures ensure to find dominant routes that are significantly persistent. On the negative side, RPL nodes may suffer from severe unreliability problems, mainly because RPL lacks of a complete knowledge of link qualities and it may sometimes select sub-optimal paths with highly unreliable links. Future work is also required to investigate the impact of varying traffic loads, channel conditions and different traffic patterns on the RPL stability and reliability.

As a final remark, we believe that the insights gained in this paper can facilitate the design of new mechanisms to improve the reliability and adaptability of RPL. For instance, we plan to investigate more sophisticated hysteresis techniques that mitigate the risk of using low-quality links. Furthermore, lightweight channel probing techniques should be integrated in RPL to improve routing efficiency. We want also to explore how routing reliability can improve with the use of more advanced routing schemes, such as multi-path and network coding.

References

[1] N. Accettura, L. Grieco, G. Boggia, and P. Camarda. Performance analysis of the rpl routing protocol. In *Proc. of IEEE ICM'11*, pages 767–772, 2011.

[2] P. Asmus. Microgrids, virtual power plants and our distributed energy future. *The Electricity Journal*, 23(10):72–82, 2010.

[3] C. Boano, K. Romer, F. Osterlind, and T. Voigt. Realistic Simulation of Radio Interference in COOJA. In *Proc. of EWSN'11*, February 2011.

[4] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi. The deployment of a smart monitoring system using wireless sensor and actuator networks. In *Proc. of IEEE SmartGridComm'10*, pages 49–54, 2010.

[5] T. Clausen and U. Herberg. Comparative study of RPL-enabled optimized broadcast in Wireless Sensor Networks. In *Proc. of IEEE ISSNIP'10*, pages 7–12, 2010.

[6] T. Clausen, U. Herberg, and M. Philipp. A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL). In *Proc. of IEEE WiMob'11*, pages 365–372, 2011.

[7] M. Daoud and X. Fernando. On the Communication Requirements for the Smart Grid. *Energy and Power Engineering*, 3:53–60, 2011.

[8] M. Dohler, T. Watteyne, T. Winter, and D. Barthel. Routing Requirements for Urban Low-Power and Lossy Networks. IETF RFC 5548, 2009.

[9] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. In *ACM SIGCOMM'04*, pages 133–144, Aug. 30—Sept. 3 2004.

[10] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proc. of IEEE LCN'04*, pages 455–462, 2004.

[11] H. Farhangi. The path of the smart grid. *IEEE Power and Energy Magazine*, 8(1):18–28, 2010.

[12] S. Galli, A. Scaglione, and Z. Wang. For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid. *Proceedings of the IEEE*, 99(6):998–1027, 2011.

[13] T. Garrity. Getting smart. *IEEE Power and Energy Magazine*, 6(2):38–45, 2008.

[14] O. Gnawali and P. Levis. The Minimum Rank with Hysteresis Objective Functions. Internet Draft v07, March 2012.

[15] V. Gungor, B. Lu, and G. Hancke. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *IEEE Transactions on Industrial Electronics*, 57(10):3557–3564, 2010.

[16] D. Hart. Using ami to realize the smart grid. In *Proc. of IEEE Power and Energy Society General Meeting'08*, 2008.

[17] U. Herberg and T. Clausen. A comparative performance study of the routing protocols LOAD and RPL with bi-directional traffic in low-power and lossy networks (LLN). In *Proc. of ACM PE-WASUN '11*, pages 73–80, 2011.

[18] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. IETF RFC 6282, September 2011.

[19] P. Kulkarni, S. Gormus, Z. Fan, and B. Motz. A self-

organising mesh networking solution based on enhanced RPL for smart metering communications. *Proc. of IEEE HotMESH'11*, 2011.

[20] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The Trickle Algorithm. IETF RFC 6206, March 2011.

[21] J. Martocci, P. De Mil, N. Riou, and W. Vermeylen. Building Automation Routing Requirements in Low-Power and Lossy Networks. IETF RFC 5867, 2010.

[22] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). IETF RFC 4861, September 2007.

[23] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-Level Sensor Network Simulation with COOJA. In *Proc. of IEEE LCN'06*, pages 641–648, 2006.

[24] B. Pavković, F. Theoleyre, and A. Duda. Multipath opportunistic RPL routing over IEEE 802.15.4. In *Proc. of ACM MSWiM'11*, pages 179–186, 2011.

[25] D. Popa, J. Jetcheva, N. Dejean, R. Slazar, and K. Hui, J. amd Monden. Applicability Statement for the Routing Protocol for Low Power and Lossy Networks (RPL) in AMI Networks. Internet Draft, 2011.

[26] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth. Routing stability in static wireless mesh networks. In *Proc. of ACM PAM'07*, pages 73–83, 2007.

[27] E. Santacana, G. Rackliffe, L. Tang, and X. Feng. Getting smart. *IEEE Power and Energy Magazine*, 8(2):41–48, 2010.

[28] K. Spees and L. Lave. Demand response and electricity market efficiency. *The Electricity Journal*, 20(3):69–85, 2007.

[29] P. Thubert. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). IETF RFC 6552, March 2012.

[30] J. Tripathi, J. de Oliveira, and J. Vasseur. Applicability Study of RPL with Local Repair in Smart Grid Substation Networks. In *Proc. of IEEE SmartGridComm'10*, pages 262–267, 2010.

[31] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel. Routing Metrics used for Path Calculation in Low Power and Lossy Networks. IETF RFC 6551, March 2012.

[32] D. Wang, Z. Tao, J. Zhang, and A. Abouzeid. RPL Based Routing for Advanced Metering Infrastructure in Smart Grid. In *Proc. of IEEE ICC'10*, 2010.

[33] W. Wang, Y. Xu, and M. Khanna. A survey on the communication architectures in smart grid. *Computer Networks*, 55:3604—3629, 2011.

[34] T. Watteyne, A. Molinaro, M. Richichi, and M. Dohler. From MANET To IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols. *IEEE communications Surveys & Tutorials*, 13(4):688–707, 2011.

[35] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550, March 2012.

[36] G. Xiong, C. Chen, S. Kishore, and A. Yener. Smart (in-home) power scheduling for demand response on the smart grid. In *Proc. of IEEE PES ISGT'11*, 2011.

[37] S. Yeleti and Y. Fu. Impacts of energy storage on the future power system. In *Proc. of IEEE NAPS'10*, 2010.