RPL at Scale: Experiences from a Performance Evaluation on up to 700 IEEE 802.15.4 Devices

Mateusz Banaszek*, Markus Schuß†, Carlo Alberto Boano†, and Konrad Iwanicki*
*Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
†Institute of Technical Informatics, Graz University of Technology, Austria
E-mails: {m.banaszek, iwanicki}@mimuw.edu.pl; {markus.schuss, cboano}@tugraz.at

***Abstract*—The scalability of the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is increasingly attracting the interest of both academia and industry. In light of this, we present experiences from a pilot experimental study conducted over several weeks on the protocol's state-of-the-art implementation, RPL-Lite, in a large multi-hop IEEE 802.15.4 network of up to 700 nodes dispersed over 3390 m^2. Our results show that RPL is capable of reliable and self-managed data collection even in such large-scale deployments, but its performance may drop significantly because of transmission power settings, border router locations, and insufficient load balancing, notably due to a phenomenon that, to the best of our knowledge, has not been reported in literature. We believe that our observations can be of value to designers and administrators of low-power wireless networks as well as to the community developing RPL.**

***Index Terms*—Contiki-NG, Internet of Things, Low-power wireless, Multi-hop sensor networks, RPL, 1KT testbed.**

I. Introduction

Since its standardization by the IETF over a decade ago, the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [1] has gained significant adoption in industry. By centralizing parts of the route selection and maintenance functionality from resource-constrained low-power wireless devices to more powerful border routers and incorporating multiple self-regulation mechanisms, the protocol—even when operating over inherently unreliable low-power wireless links—is able to offer end-to-end packet delivery rates as high as 99.999% [2] while substantially facilitating network management.

Because of its industrial impact, RPL has received considerable research attention. Previous work concerned, among others, interoperability, reliability, cross-layer behavior, and effects of mobility (see surveys by Kharrufa et al. [3] and Darabkh et al. [4]), thereby contributing to the protocol's performance and usability. Today, in turn, since RPL is relatively mature, the focus of the research community is shifting to analyzing the protocol's behavior in *large and dense networks*, that is, with hundreds of participating nodes [5], [6].

One of the reasons for this shift is RPL's approach to scalability and its practical implications. Because of the partial centralization of routing, expanding a network with a number of low-power wireless nodes is expected to also require extra border routers to handle those devices. However, in many real-world applications, installing a border router at a new location is problematic, as it requires provisioning a tethered power supply and Internet access. From a management perspective, it is thus beneficial to minimize the number of border routers in a network.[1] Therefore, information on how well a single border router can handle a network of a given size and what problems may occur when scaling up the number of nodes is crucial, among others, for planning real-world deployments.

The gap to fill. A large body of existing work studying RPL's performance at scale employs simulation [6], [7], [8], [9]. Because of the reproducibility and tracing features offered by simulators, such approaches are invaluable for identifying various issues. However, simulators do not model all real-world phenomena, especially those that are computationally too complex or of which we are yet unaware. Therefore, for benchmarking RPL's performance, in addition to simulators, researchers conduct real-world experiments.

To date, however, large-scale real-world experimental studies on RPL have focused primarily on *downward routing*, from a border router to low-power devices (e.g., involving a 352-node testbed [2] or a 500+-node deployment [5]). In contrast, up-to-date insights into the scalability of *upward routing*, from low-power devices to a border router, are scarce. Although upward routing was studied extensively in the past, those works involved far smaller scales and are relatively outdated: they employed now-vintage low-power wireless devices (e.g., TelosB), and typically only tens of them [10], [11], [12], with the largest published study [13] (251 devices) dating back to 2015, when RPL's implementations were not yet mature.

However, benchmarking the large-scale real-world performance of upward routing is important, because this type of routing is used not only by the data plane, typically for collecting readings from sensor devices [14], [15], but also by the control plane, for gathering diagnostic information on the network by the border router or another centralized controller [16]. What is more, while the principles behind upward routing may sound simpler than for downward routing (as each node "just" needs to keep track of one parent node toward a border router and forward all messages there[2]), a massive scale and the limited knowledge of the rest of the

[1] A low number of border routers does not necessarily affect fault tolerance, especially since RPL supports replicated border routers, which synchronize their state, acting as each other's backups. Replicated routers, however, do not require distinct locations, so they can be largely treated as one (virtual) router, for which the aforementioned problems need to be solved only once.

[2] In a nutshell, RPL organizes nodes into a tree rooted at a border router. A node corresponding to a low-power device joins the tree by selecting a parent among the nodes within its radio range. This is done based on the nodes' ranks, which reflect their logical distance to the root and are advertised regularly. Rank calculation and parent selection are driven by an objective function (OF).

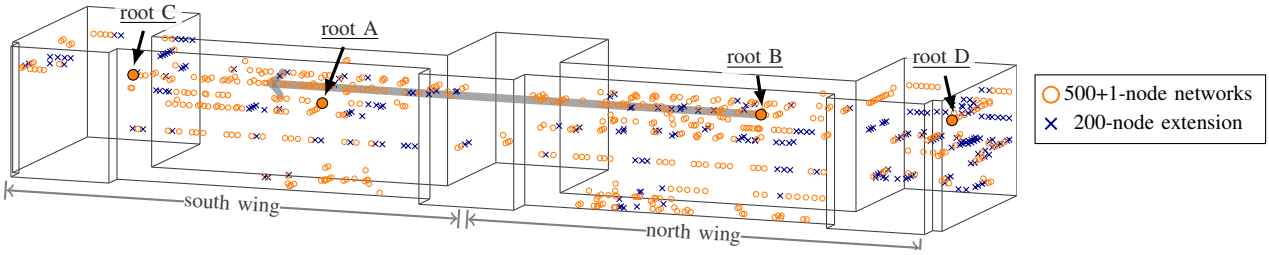


Fig. 1. Overview of the 500+1-node networks used in our evaluation, and the 200-node extension employed in Sect. III-F. Devices pointed by arrows are used as roots (one at a time). The gray arrow visualizes the link discussed in Sect. III-C (as if it were line-of-sight). The devices are deployed throughout the entire building for a total area of 3390 m^2 with its north and south wings containing, respectively, 204+62 and 297+138 (base networks+extension) devices.

network that any resource-constrained node has in upward routing may hamper RPL's performance. For example, high traffic loads or poor-quality links may result in significant rates of local frame retransmissions, thereby overloading specific nodes and causing a major message loss globally.

Our contributions. In this paper, we thus present experiences from a 500- and 700-node pilot study focused on RPL's upward routing in typical data collection settings. To this end, we leverage the 1KT testbed [17], which comprises 1000 low-power wireless devices (the off-the-shelf TI CC2650 SoCs) and allows for precisely controlling message transmissions.

As RPL's implementation, we employ the widely popular state-of-the-art *RPL-Lite* from Contiki-NG [18], which is the modern successor to the original ContikiRPL [19], one of the prototypes for the IETF standard. We run an experimental evaluation with the protocol for several weeks, during which we benchmark its performance across several dimensions. In particular, we investigate how it behaves as a function of data traffic, network density, and border router location.

Our methodology aims to avoid any bias due to the environment (in terms of performance fluctuations across day/night and weekdays/weekends), as well as due to the other layers of the protocol stack. In fact, most of the prior works evaluating RPL's performance on larger testbeds do not focus exclusively on RPL, but investigate selected network stacks with layers whose mutual behavior is heavily intertwined. For example, in [10], RPL builds on top of a MAC layer implementing TSCH and relying on Orchestra for the scheduling of slots. However, Orchestra also utilizes information from RPL. In effect, it becomes difficult to isolate the root cause of problems when performance degrades. We thus run RPL on top of only a simple always-on CSMA protocol, which also helps to avoid any dependencies on specific radio duty-cycling features.

Our experiments show that, by and large, RPL can remain stable and operate reliably, thereby requiring virtually no interventions from network administrators. With just one border router handling 500 or even 700 low-power wireless devices, RPL's upward routing is able to achieve a global end-to-end packet reception ratio (PRR) of almost 100%, despite the challenging environment of the 1KT testbed. However, we do observe problems when the traffic increases. We trace them to insufficient load balancing in RPL and identify their surprising cause, which, to the best of our knowledge, has not been reported to date in the literature. We also highlight other issues, related to border router location and radio transmission power selection, which are important from the network design and maintenance perspective.

II. Experimental Evaluation

To perform the experimental evaluation we first design a proper setup and devise a comprehensive methodology.

A. Experimental Setup

Testbed facility. We run our evaluation on 1KT [17], a single-site, 1000-node testbed deployed throughout a 5-story university building and built out of devices located directly in human spaces (e.g., under windowsills, behind desks, on file cabinets). 1KT employs a single type of devices, and hence enables an unbiased evaluation, as all nodes use the same hardware and software. A device under test (DuT) is the Texas Instruments CC2650 system-on-chip featuring an ARM Cortex-M3 CPU with 20 kB of RAM, 128 kB of flash memory, and a built-in low-power transceiver supporting IEEE 802.15.4. Each DuT is directly attached to a single-board computer that constitutes the testbed backbone: it communicates with a central server over Ethernet or Wi-Fi, flashes the DuT, forwards server commands to the DuT over UART lines, and uploads to the server log data emitted by the DuT.

Experiments execution. For most of the experiments, we select 500 devices of 1KT—thereby matching the size of the largest study on downward routing [5]—to act as *client nodes*, which generate data traffic at regular intervals, and one device to act as the *root node*, which operates as the border router, collecting all messages sent by the clients (see Fig. 1). To exercise RPL's scalability even further, in the final experiments, we employ 200 additional devices. To avoid generating Wi-Fi traffic affecting our measurements (as both Wi-Fi and IEEE 802.15.4 share the 2.4 GHz band), 1KT's active devices are connected to the central server via Ethernet. We configure the server to trigger transmissions of individual messages by sending commands to DuTs acting as the client nodes to let the network sustain a specific traffic load (e.g., 10 msg/s), and randomize the order in which nodes send messages.[3] Each node logs transmissions and receptions of messages (including those being just forwarded), key low-level

[3]For example, for a traffic load of 10 msg/s, the server sequentially instructs all 500 client nodes to send a message within 50 seconds. The order in which the nodes send their messages is randomized in consecutive 50 s rounds.

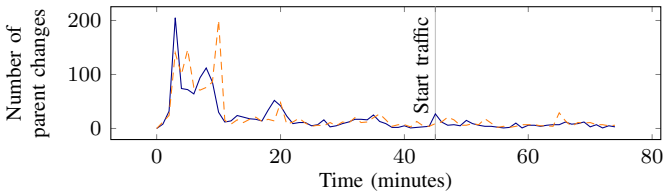


Fig. 2. In all our experiments, we allocate 45 minutes for RPL to establish routes before generating traffic. This is sufficient, as shown by the number of parent switches over time in two exemplary runs (solid blue and dashed orange lines), which consistently stabilize after ≈25 mins.

metrics, such as the link quality reported by the protocol and the changes of its neighbor table, as well as events in the routing layer, such as changes of the node's rank and parent.

Contiki-NG application firmware. DuTs execute a custom Contiki-NG[4] application running the default *RPL-Lite* stack. We keep all default settings, but disable frame fragmentation and downward routing to lower RAM and ROM usage (these features are not needed in our setup). Unless differently stated, we configure DuTs to transmit at maximum power (+5 dBm). We use IEEE 802.15.4 channel 26, as it does not overlap with commonly used Wi-Fi channels and is not used by any other network in the building. As MAC layer we employ the always-on CSMA protocol not to create any cross-layer dependencies and to avoid synchronization-related issues of duty cycling.

B. Methodology

Duration and time of experiments. Experimental studies have shown that the performance of low-power wireless networks follows periodic fluctuations due to the dynamicity of environments they are deployed in [20], [21], [17]. For example, in office environments it is common to observe a higher performance during nights and weekends when the office is at its quietest and the amount of RF interference from surrounding devices is limited. As the building in which 1KT is deployed hosts students, academics, and administrative staff, it is a fairly busy place during weekdays, but relatively quiet at nights and weekends. For this reason, we differentiate between experiments run over weekends and weekdays to *account for the dynamicity of the environment*. Moreover, we make comparisons only between results obtained during the same academic period (classes session, exam session, etc.). Finally, we run experiments on a daily basis, so that we can capture fluctuations along the 24 hours. Specifically, we start each experiment at 00:45 by instructing the server to flash all nodes with our firmware. We then wait until 01:30 to let RPL build routes in absence of network traffic (see Fig. 2), and then we instruct the server to initiate the generation of messages at each client following a given traffic load. We generate the traffic for 22 hours and 45 minutes. We then keep a 30-minute buffer, during which we disable all nodes to ensure that no old message is generated when the new experiment starts.

Explored dimensions. We vary *the traffic load in the network* from 10 to 90 msg/s, which results in 819,000 up to 7,371,000 messages being sent by client nodes per experiment. We fix the payload size of the messages to 6 bytes: this resembles an application regularly collecting simple sensor data across a large-scale network. As the nodes in 1KT are deployed throughout a large building, we investigate RPL's performance when selecting *different nodes as root node*. We specifically pick two devices at the periphery of the building (nodes marked as root `C` and root `D` in Fig. 1) and two devices in the middle of the building (nodes marked as root `A` and root `B` in Fig. 1). We further investigate whether RPL's performance varies when decreasing *the density of the network*. To this end, we decrease the TX power from +5 dBm to +1 dBm.

Metrics of interest. We are primarily interested in the reliability of communications. We hence compute the Packet Reception Ratio (*PRR*) across the network as the ratio of messages sent by all client nodes that were correctly received at the root node, and we report both the average across the entire experiments, as well as the statistics split in 15-minute intervals. We further compute the number of *TX attempts* required to successfully transmit a frame as reported by the MAC layer, the average and maximum number of *hops* between clients and root, as well as the number of *parent changes* (calculated per hour per node) and the average Expected Transmission Count (ETX) as an estimation of the link quality for all child-to-parent links (*ETX to parent*). For each of these metrics (except for the number of parent changes), we report the minimum, median, and maximum value over 15-min intervals excluding the largest and the smallest outlier.

III. Evaluation Results

For each explored dimension we run a series of experiments and calculate the metrics shown in Tab. I. We report the results of 19 experiments lasting 21 days in total (one per scenario).

A. RPL Performance at Scale

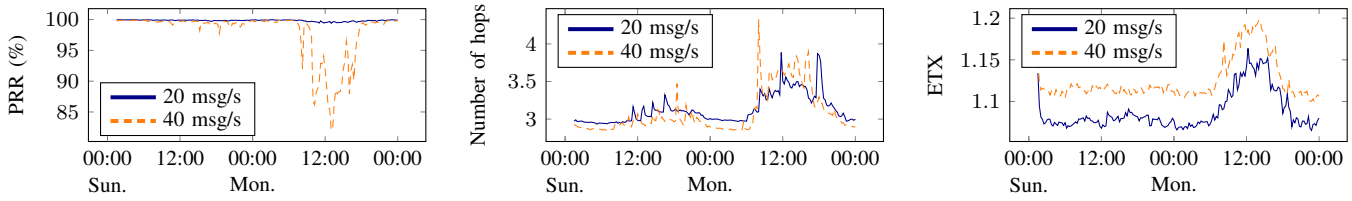
We first validate our setup by studying the performance of RPL with a moderate traffic load (20 and 40 msg/s) when using as the root node the device marked as root `A` in Fig. 1. We run each experiment on two consecutive days (doubling 24-hour runs): a Sunday (weekend), and a Monday (weekday), in order to also assess the severity of changes in the environment.

Despite the large scale, with the traffic load of 20 msg/s the network sustains an average PRR close to 100% at all times, without significant differences across day/night or weekday/weekend (see Fig. 3a). When increasing the traffic load to 40 msg/s, an only slightly lower PRR is observed at night and on Sunday. However, during daytime on Monday the median and minimum PRR drops noticeably to 92.9% and 82.4%, respectively, highlighting the impact that the presence of people has on low-power wireless communication in office environments. The impact is particularly visible also in the difference in average number of hops (see Fig. 3b), and average ETX to parent (see Fig. 3b). During Monday daytime, we record also an increase in the number of parent changes (≈+40–70%), and in the number of attempts required at the MAC layer to successfully transmit a frame (≈+5%).

[4]We employ the—at the time of experimentation—latest stable release (v4.8) of Contiki-NG OS [18] extended with support for 1KT.

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS CHARACTERIZING RPL'S PERFORMANCE AT THE SCALE OF 500 AND 700 (THE BOTTOM PART) NODES

Explored dimension			PRR (%)	PRR (%) 15-min interval min–median–max	TX attempts 15-min interval min–median–max	Avg. hops 15-min interval min–median–max	Max. hops 15-min interval min–median–max	Parent changes (#/hour/node)	ETX to parent 15-min interval min–median–max
RPL's performance broken down per day of the week and time of the day for traffic loads of 20 msg/s and 40 msg/s (during classes session).									
20 msg/s	Sun	day	99.88	99.7–99.9–99.9	1.09–1.10–1.11	3.0–3.0–3.3	6–6–7	1.0	1.07–1.08–1.09
		night	99.90	98.1–99.9–100	1.09–1.10-1.12	2.9–3.0–3.1	6–6–7	0.9	1.07–1.07–1.09
	Mon	day	99.66	99.4–99.7–99.9	1.09–1.16–1.22	3.1–3.4–3.9	7–7–9	1.4	1.08–1.12–1.15
		night	99.90	99.8–99.9–100	1.09–1.09–1.11	3.0–3.0–3.2	6–7–8	1.0	1.07–1.07–1.09
40 msg/s	Sun	day	99.50	98.2–99.6–99.8	1.14–1.15–1.16	2.9–3.0–3.1	5–6–7	1.0	1.11–1.11–1.12
		night	99.63	98.0–99.8–99.9	1.14–1.15–1.17	2.9–2.9–3.1	5–6–7	0.9	1.11–1.11–1.12
	Mon	day	93.08	82.4–92.9–99.8	1.14–1.21–1.30	3.1–3.6–3.9	7–8–11	1.7	1.11–1.16–1.20
		night	99.67	99.2–99.7–99.9	1.14–1.16–1.18	2.9–2.9–3.1	5–6–7	1.1	1.10–1.11–1.12
RPL's performance broken down per day of the week and time of the day for the north and the south wing (traffic load: 40 msg/s, classes session).									
South	Sun	day	99.97	100–100–100	1.12–1.14–1.15	1.6–1.7–1.7	4–4–4	0.7	1.11–1.12–1.13
		night	99.98	100–100–100	1.12–1.13–1.14	1.6–1.7–1.7	3–4–4	0.7	1.11–1.12–1.13
	Mon	day	99.89	99.7–99.9–100	1.14–1.19–1.29	1.7–1.9–2.1	4–4–6	1.0	1.11–1.14–1.19
		night	99.98	100–100–100	1.11–1.14–1.16	1.6–1.7–1.8	3–4–4	0.8	1.10–1.11–1.12
North	Sun	day	99.18	97.1–99.4–99.7	1.14–1.17–1.18	3.7–3.9–4.1	5–6–7	1.2	1.10–1.11–1.12
		night	99.39	98.0–99.7–99.8	1.15–1.17–1.19	3.7–3.7–4.1	5–6–7	1.1	1.10–1.11–1.12
	Mon	day	88.41	70.4–88.0–99.6	1.13–1.21–1.32	4.0–4.6–5.2	7–8–11	2.3	1.11–1.18–1.21
		night	99.46	98.6–99.6–99.8	1.15–1.18–1.22	3.7–3.7–4.0	5–6–7	1.2	1.10–1.11–1.12
RPL's performance during weekdays (classes session) as a function of traffic load varying between 10 msg/s and 90 msg/s.									
10 msg/s			98.24	71.4–99.9–100	1.05–1.05–1.20	3.0–3.9–5.8	5–8–18	1.8	1.05–1.08–1.21
20 msg/s			96.61	63.8–99.7–100	1.08–1.11–1.20	3.0–4.0–5.2	5–9–15	2.6	1.07–1.10–1.21
30 msg/s			98.75	93.7–99.6–99.9	1.12–1.15–1.24	2.9–3.3–5.5	6–7–13	1.6	1.09–1.11–1.19
40 msg/s			93.48	80.9–95.7–99.7	1.15–1.18–1.27	2.9–3.6–4.4	6–8–11	1.4	1.11–1.13–1.18
50 msg/s			89.14	67.4–91.9–99.1	1.16–1.20–1.25	3.1–4.1–5.1	6–9–13	2.1	1.12–1.15–1.22
70 msg/s			71.60	57.9–71.0–86.5	1.21–1.24–1.29	3.2–4.3–5.4	6–8–13	1.8	1.14–1.16–1.21
90 msg/s			60.96	50.0–61.7–68.3	1.21–1.27–1.32	3.0–3.7–5.7	6–8–13	2.5	1.17–1.20–1.27
RPL's performance during weekdays (exam session) as a function of the root node location (traffic load: 40 msg/s).									
root A			95.59	82.1–97.5–99.7	1.13–1.16–1.24	3.0–3.2–4.2	5–7–9	1.2	1.11–1.13–1.18
root B			98.09	93.9–98.7–99.9	1.16–1.19–1.23	2.6–3.0–4.0	5–6–11	1.4	1.11–1.13–1.18
root C			86.98	75.5–88.0–94.0	1.17–1.19–1.23	3.5–3.7–5.2	6–7–11	1.5	1.12–1.14–1.20
root D			71.85	53.7–69.8–95.0	1.17–1.20–1.26	4.9–5.8–7.0	8–11–13	1.9	1.14–1.16–1.20
RPL's performance during weekdays (exam session) and weekend as a function of network density (traffic load: 40 msg/s).									
Weekend	+5 dBm		99.56	98.2–99.7–99.9	1.14–1.15–1.17	2.9–2.9–3.2	5–6–8	1.0	1.11–1.11–1.12
	+1 dBm		89.29	66.4–90.3–97.3	1.16–1.21–1.25	3.4–4.1–6.6	8–9-19	2.4	1.12–1.14–1.17
Weekday	+5 dBm		95.59	82.1–97.5–99.7	1.13–1.16–1.24	3.0–3.2–4.2	5–7–9	1.2	1.11–1.13–1.18
	+1 dBm		86.69	53.5–89.4–98.6	1.16–1.20–1.24	2.7–3.7–8.3	7–8–22	2.4	1.11–1.13–1.20
RPL's performance with 700 nodes during weekdays (summer holiday) as a function of traffic load varying between 30 msg/s and 50 msg/s.									
30 msg/s			99.82	99.4–99.8–99.9	1.13–1.14–1.19	2.8–2.9–3.4	5–5–9	1.2	1.09–1.10–1.13
40 msg/s			99.03	93.6–99.4–99.7	1.16–1.18–1.21	2.8–3.1–3.7	6–6–10	1.5	1.11–1.12–1.17
50 msg/s			95.63	90.0–95.8–99.5	1.18–1.20–1.23	3.0–3.4–4.0	6–7–11	1.5	1.13–1.14–1.16



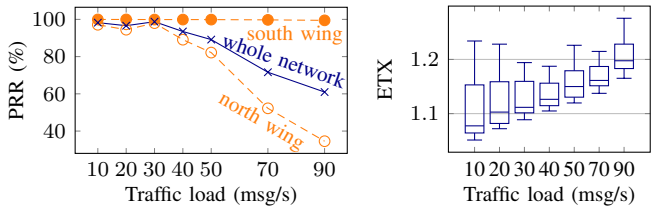
(a) PRR (b) Avg. number of hops to the root (c) Avg. child-to-parent ETX

Fig. 3. Metrics calculated over 15-minute intervals for experiments with a traffic load of 20 msg/s and 40 msg/s.

Performance in north and south wings. We analyze whether RPL's performance is homogeneous across the building in which the testbed is located. We notice a significant disparity in performance sustained by the nodes located in the north and in the south wing of the building (see Fig. 1) when transmitting 40 msg/s to root A. For the south wing, the PRR is nearly always 100%, and all other metrics show minimal differences over time (except for a slight worsening during Monday daytime). On the contrary, for the north wing, compared to the south wing, the PRR temporarily decreases by as much as 2.9% on Sunday and 29.3% on Monday during daytime. The north wing exhibits also a higher number of parent changes (up to +130%). We hence conclude that the performance throughout the network is largely dishomogeneous.

B. Impact of Traffic Load

We run a series of 24-hour experiments with traffic loads from 10 to 90 msg/s, executing all experiments only during



(a) Avg. PRR (b) Avg. child-to-parent ETX

Fig. 4. Metrics calculated for experiments transmitting 10 msg/s to 90 msg/s.

weekdays to avoid biased comparisons. For traffic loads up to 30 msg/s, the network sustains high PRR, although it can experience temporary PRR drops, and a high rate of parent changes. With higher traffic loads, the average PRR drops noticeably: below 90% when sending 50 msg/s, and down to 61% when sending 90 msg/s (see Fig. 4a). We observe a correlation across the average ETX to parent and the traffic load (see Fig. 4b), which reflects an increasing congestion of the network. The average and maximal numbers of hops, instead, do not seem correlated to the traffic load (not plotted).

A breakdown of the PRR sustained in the individual wings of the building (see Fig. 4a) is in line with the results shown in Sect. III-A, and highlights the north wing exhibiting a worse performance than the south wing.

C. The Need for Load Balancing

To understand the poor performance of the north wing, we reconstruct actual routes taken by each message, and focus on the experiment with a traffic load of 50 msg/s. 63% of messages that do not reach the root node are received by device with ID *1e8* (marked root `B` in Fig. 1 as it acts as a root node in another experiment), but are not forwarded any further. This node reports only few unsuccessful transmissions, but it simply does not manage to send messages as fast as it receives them, leading to overflows in the MAC layer's transmit queue (the node drops up to 80% of the messages it should forward).

When analyzing a snapshot of the routes established during daytime (see Fig. 5a), we can observe that node *1e8* is chosen as preferred parent by several children because of its short path to the root node[5], which explains why it receives so many messages. Node *1e8* does not, however, represent a physical bottleneck, as all its children would have valid alternative routes to the root node. The route via node *1e8* encompasses only a few hops, because the link between *1e8* and its parent is exceptionally long: the physical distance between the two devices is nearly 80 meters, which is much more than that of other links (see Fig. 5b). While one may argue that this link belongs to the *gray area* in low-power wireless communication [22], which comprises long links that exhibit erratic PRR values, the link actually displays consistently reasonable performance. In either case, although existing literature lists several phenomena causing unbalanced traffic in RPL [23], to the best of our knowledge, no work has reported on the adversarial impact of physically exceptionally-long but still relatively-good-quality links. This observation reinforces the importance of real-world experiments with RPL in truly large networks. A higher PRR during nighttime than daytime we partially attribute to more long, good-quality links, and more balanced traffic we observe then.

Introducing load balancing. Unbalanced traffic in RPL can be mitigated, for instance, by introducing OFs which account for the queue utilization to avoid overflows [24] (e.g., QU-RPL [25]). We modify RPL Lite's default OF such that it penalizes dropping messages due to overflows by increasing the node's rank proportionally (a simple reactive approach compared to QU-RPL's proactive approach). Already this unsophisticated scheme reduces significantly the number of overflows (≈ -35% for 50 msg/s) and increases PRR (≈ +7%) in the north wing. We hence conclude that unbalanced routes are indeed the major performance limitation, and that a full-featured load balancing can substantially increase network performance. However, such mechanisms are not explicitly addressed by the standard and are thus not widely implemented.

D. Impact of Root Location

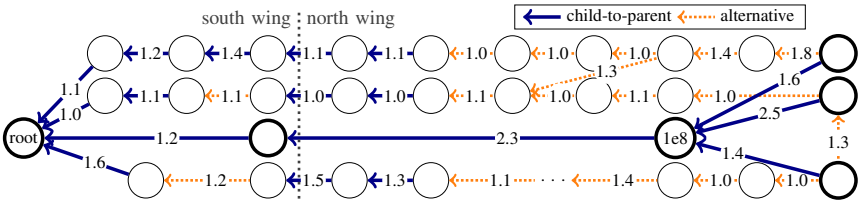
All experiments reported hitherto employ the root node in a dense region of the south wing (root `A`, see Fig. 1). To study the impact of root location on the overall performance of the network, we select three other devices to act as root nodes: one located in the middle of the north wing (root `B`), and two located at the periphery of the network (root `C` and `D`).

Physical locations of the root nodes are reflected in their logical positions within the network, as the client nodes establish short routes to root `A` and root `B`, longer routes to root `C`, and noticeably the longest routes to root `D`. When root nodes are located in the middle of the network, the network sustains a high average PRR (98% with root `B`, and 96% with root `A`). Deploying root nodes at the periphery of the network, on the other hand, results in a PRR lower by 11% (root `C`) and 26% (root `D`). We observe also noticeable temporary drops in the PRR when employing these root nodes.

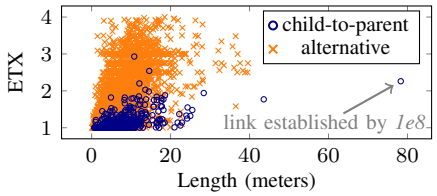
E. Impact of Network Density

We reduce the TX power from +5 dBm to +1 dBm, and compare the resulting networks transmitting 40 msg/s during a weekend and on a weekday. The power reduction results, as expected, in more hops to the sink. However, it also causes a sizable decrease of the average PRR: by 10% during the weekend, and by 9% on weekdays. The decrease in performance is also confirmed by significant temporary drops in the PRR (down to 53.5% on weekdays), and worsened quality of most routes (not reported in Tab. I): when transmitting at +5 dBm, all nodes have good connectivity to the sink (PRR 90%–100%), but when the power is reduced, nearly half of the nodes have only a discrete connectivity (PRR 10%–90%). A lower TX power results also in more than twice the rate of parent changes, even if we do not observe significantly worse child-to-parent ETX during weekdays.

[5]Contiki-NG implements two OFs, MRHOF and OF0, which select the route based not only on its length, but also on the quality of its links (ETX). The very short route via *1e8*, however, outbalances the impact of lower-quality links it traverses. None of the two OFs accounts for queue overflows.



(a) The route to the root node via node *1e8* and alternative routes of *1e8*'s selected children. The numbers on arrows refer to the ETX of each link.



(b) ETX and length of links as if they were line-of-sight. Only links of ETX ≤ 4 are plotted.

Fig. 5. A partial snapshot of the network topology (child-to-parent and alternative links) recorded in our experiments.

F. Beyond 500 nodes

Our results suggest that the network size itself is not a limiting factor, and that with the preferable root node location and the high TX power we should observe a PRR close to 100% also after extending the network with more devices.

Indeed, a network with 700 client nodes (the total number of 1KT's devices available to us, see Fig. 1) and root B, during a limited people's presence in the building (summer holiday), sustains an average PRR over 99% for traffic loads up to 40 msg/s, and show no anomalies in other metrics.

IV. Conclusions and Outlook

Our pilot study shows that RPL is capable of reliable and largely self-managed upward routing at the scale of 700 low-power wireless devices with just 1 border router. However, there are a few pitfalls, pertaining, among others, to the selection of border router location and transmission power, that may result in an underperforming network, thereby requiring attention from its architects and managers. Especially one discovered phenomenon—the impact on load balancing of exceptionally long, albeit relatively good-quality links—may also necessitate extensions or improvements to the protocol itself, or at least exposing appropriate management mechanisms that would allow for manually altering RPL's behavior in this respect at runtime. Overall, we believe that our experiences can be of value to designers and administrators of low-power wireless networks as well as the community developing RPL.

Acknowledgments

The authors would like to thank Maciej Matraszek and Wojciech Ciszewski for their input. This work was supported by the National Science Center (NCN) in Poland under grant no. 2019/33/B/ST6/00448. The first author was also supported by the IDUB Programme (agreement no. 01/IDUB/2019/04).

References

[1] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," RFC 6550, Internet Engineering Task Force, 2012.

[2] S. Duquennoy, J. Eriksson, and T. Voigt, "Five-Nines Reliable Downward Routing in RPL," *Cornell University – arXiv:1710.02324*, 2017.

[3] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, "RPL-based routing protocols in IoT applications: A review," *IEEE Sensors J.*, vol. 19, no. 15, 2019.

[4] K. A. Darabkh, M. Al-Akhras, J. N. Zomot, and M. Atiquzzaman, "RPL Routing Protocol over IoT: A Comprehensive Survey, Recent Advances, Insights, Bibliometric Analysis, Recommendations, and Future Directions," *J. Netw. Comput. Appl.*, vol. 207, 2022.

[5] J. Eriksson, N. Finne, N. Tsiftes, S. Duquennoy, and T. Voigt, "Scaling RPL to Dense and Large Networks with Constrained Memory," in *Proc. 15th EWSN Conf.*, 2018.

[6] S. A. Abdel Hakeem, A. A. Hady, and H. Kim, "RPL Routing Protocol Performance in Smart Grid Applications Based Wireless Sensors: Experimental and Simulated Analysis," *Electronics*, vol. 8, no. 2, 2019.

[7] A. Arena, P. Perazzo, C. Vallati, G. Dini, and G. Anastasi, "Evaluating and improving the scalability of RPL security in the Internet of Things," *Comput. Commun.*, vol. 151, 2020.

[8] X. Liu, Z. Sheng, C. Yin, F. Ali, and D. Roggen, "Performance Analysis of Routing Protocol for Low Power and Lossy Networks (RPL) in Large Scale Networks," *IEEE Internet Things J.*, vol. 4, no. 6, 2017.

[9] K. S. Bhandari, I.-H. Ra, and G. Cho, "Multi-topology based QoS-differentiation in RPL for Internet of Things applications," *IEEE Access*, vol. 8, 2020.

[10] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks through Autonomously Scheduled TSCH," in *Proc. 13th SenSys Conf.*, 2015.

[11] E. Ancillotti, R. Bruno, and M. Conti, "Reliable Data Delivery with the IETF Routing Protocol for Low-Power and Lossy Networks," *IEEE Trans. Ind. Inform.*, vol. 10, no. 3, 2014.

[12] S. Dawans, S. Duquennoy, and O. Bonaventure, "On Link Estimation in Dense RPL Deployments," in *Proc. 7th SenseApp Workshop*, 2012.

[13] J. Isern, A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "Large-Scale Performance Evaluation of the IETF Internet of Things Protocol Suite for Smart City Solutions," in *Proc. 12th PE-WASUN Symp.*, 2015.

[14] S. Atalla *et al.*, "IoT-enabled precision agriculture: Developing an ecosystem for optimized crop management," *Information*, vol. 14, no. 4, 2023.

[15] I. AlShiab, A. Leivadeas, and M. Ibnkahla, "Virtual sensing networks and dynamic RPL-based routing for IoT sensing services," in *Proc. ICC Conf.*, 2021.

[16] T. Eckert, M. H. Behringer, and S. Bjarnason, "An autonomic control plane (ACP)," RFC 8994, Internet Engineering Task Force, 2021.

[17] M. Banaszek *et al.*, "1KT: A Low-Cost 1000-Node Low-Power Wireless IoT Testbed," in *Proc. 24th MSWiM Conf.*, 2021.

[18] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG Open Source Operating System for Next Generation IoT Devices," *SoftwareX*, vol. 18, 2022.

[19] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-Power Wireless IPv6 Routing with ContikiRPL," in *Proc. 9th IPSN Conf.*, 2010.

[20] C. A. Boano and K. Römer, "External radio interference," in *Radio Link Quality Estimation in Low-Power Wireless Networks*, 2013, Springer-Briefs Elect. Comput. Eng.

[21] R. Jacob, M. Zimmerling, C. A. Boano, L. Vanbever, and L. Thiele, "Designing Replicable Networking Experiments With Triscale," *J. Syst. Res.*, vol. 1, no. 1, 2021.

[22] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st SenSys Conf.*, 2003.

[23] D. Pancaroglu and S. Sen, "Load balancing for RPL-based Internet of Things: A review," *Ad Hoc Networks*, vol. 116, 2021.

[24] C. Lim, "A survey on congestion control for RPL-based wireless sensor networks," *Sensors*, vol. 19, no. 11, 2019.

[25] H.-S. Kim, J. Paek, and S. Bahk, "QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications," in *Proc. 12th SECON Conf.*, 2015.