Evaluation of RPL for Medium Voltage Power Line Communication

Oana Balmau[1], Dacfey Dzung[2], Abdulkadir Karaağaç[1], Vukasin Nesovic[3],
Aleksandar Paunovic[1], Yvonne Anne Pignolet[2], Niloufar Alipour Tehrani[1]
[1] EPFL, Switzerland, [2] ABB Corporate Research, Switzerland, [3] Universitaet Basel, Switzerland

Abstract—**The advantages of using power line communication (PLC) to make grids more intelligent and provide new applications are obvious: in order to distribute electricity there are power lines to all buildings and facilities, no additional cables have to be deployed and no wireless transmission power or other regulations have to be observed. However, there are currently no large communication networks using medium voltage power line communication. In this paper we evaluate the performance of the IPv6 routing protocol RPL designed for networks with low-power devices and lossy links on PLC links in a medium voltage scenario. For this purpose the embedded OS Contiki networking stack has been extended on the MAC and networking layer. To capture realistic conditions, we implemented a SINR (signal-to-noise+interference-ratio) model of the MV PLC channel for the COOJA network simulator. This enables us to investigate latency, success rate, control traffic for a partly meshed medium voltage grid simulation scenario based on the grid layout of a local utility. Thus we provide a benchmark to evaluate networking protocols for monitoring and control applications for smart grids. Based on the evaluation results, we can identify application scenarios where RPL is suitable as well as some of the systems's constraints and drawbacks.**

I. Introduction

Communication networks for smart grids are likely to consist of legacy and new communication links, using heterogeneous technologies, such as copper wires, optical fiber, wireless and power line communication [8], [12]. These networks will disseminate measurements and control commands to use synergies between different sectors, e.g., to optimize the interaction of renewable energy resources and transportation. Monitoring and control applications covering large numbers of devices need to be connected to each other.

High loss rates, low data rates and instability are specific to both wireless and power line communication (PLC) networks and require routing protocols addressing their characteristics. The IETF ROLL working group has designed the *RPL* IPv6 routing protocol that satisfies the constraints of Low Power and Lossy Networks (RFC 6550). RPL has been implemented on a number of platforms and several papers investigate it. However, to the best of our knowledge, its performance on medium voltage power lines has not been studied yet.

Carrying out experiments on medium voltage power lines requires a lot of effort and personnel. Simulations offer a valuable tool to test new protocols before deployment, avoiding significant investments, only to find out that something does not work. The main advantage of using simulations is that they ensure simple and efficient protocol development and configuration, as well as a controllable environment leading to repeatable results. On the other hand, due to simplistic implementations and model assumptions their significance and validity is limited. To avoid issues related to implementation inaccuracies and to consider the interaction between different layers, we decided to use a hybrid evaluation approach based on the cross-level emulation tool COOJA [11]. COOJA is a Java-based system simulator for networks of sensors running the Contiki operating system. Contiki OS is designed for embedded systems with memory constraints and is typically used for Internet of Things. The Contiki networking stack has been widely adopted in academia and industry. It includes modules for 6LoWPAN header compression and RPL. The current version (available on www.contiki-os.org/) supports one communication interface and offers several MAC layer implementations for wireless networks.

A power-line specific difficulty stems from transformers and circuit breakers, which may interrupt signals. To be able to communicate despite this, devices with more than one coupler offer a solution, as the couplers can be attached to the power line on either side of the primary equipment and thus allow the device to forward messages across these obstacles. This requires a communication stack that can handle this issue by a network layer with several interfaces. Thus we extended the Contiki networking stack to provide this feature, implemented the MAC layer of IEEE1901.2 and added a Signal-to-Noise-plus-Interference (SINR) model capturing how PLC signals propagate and interact with COOJA.

Using the above, we evaluate the performance of RPL for medium voltage (MV) power line networks in a realistic smart grid scenario by simulation. We used the medium voltage grid topology of a town in Switzerland, with a few primary substations as data sinks collecting measurement messages sent by ring main units. Each of the ring main units and the substations are equipped with devices that have two couplers attached. We created traffic that mimics the load of the IEC 60870-9-104 protocol by sending data periodically from the ring main units to the substations.

Summary of Contributions: We extended the COOJA simulator. In particular, we devised a SINR communication model for medium voltage power line communication and its implementation for the simulator COOJA. We extended the networking stack of Contiki to deal with multiple interfaces and implemented the MAC layer of IEEE1901.2. Apart from the simulation evaluation described in this article we deployed

978-1-4799-4934-2/14/$31.00 ©2014 IEEE

the same code base on three different platforms to confirm the functional behaviour of the stack. Our networking stack extensions allow multi-technology networks: the network layer decides which communication interface is chosen depending on the current link qualities. To evaluate the performance of communication stacks we create a benchmark scenario tailored to smart grid applications. Together with modifications of the Contiki uIP networking stack for multi-interface PLC communication (affecting the MAC layer, RPL and IP algorithms and data structures), this enables us to run simulations of a smart grid communication network and analyse its performance regarding success rate, latency, stability, etc.

Organization After a discussion of related work in Section II, we review smart grid applications and their communication requirements in Section III. Subsequently we give an overview of our networking stack extensions and the simulator in Section IV before we introduce our medium voltage power line communication model and its implementation in Section V. In Section VI we present our benchmark simulation scenario followed by an evaluation of our system in Section VII.

II. Related Work

Several aspects of the RPL routing protocol have been investigated in the past, mainly focusing on the performance of RPL on (simulated) wireless networks [18], [6], [20], [3] and low voltage PLC [3], [5], [17].

The applicability of RPL on wireless channels for Smart Grid communication networks is discussed in [1], [10], [14], [19], [22]. [17] simulated RPL for Advanced Metering (AMI) on low voltage powerline networks. Moreover, they carried out some measurements that demonstrate that their model fits the real behaviour well. However, to the best of our knowledge, the behaviour of RPL has not been investigated for applications over medium voltage power line communication.

The definition of a realistic channel model is complex for PLC and there are no well-accepted models for network level simulation of PLC. Apart from many other factors, in medium voltage networks the packet success rate is strongly dependent on the distance between the two nodes that communicate. A simple approximation assumes an exponential signal attenuation with distance [13]. Most other PLC simulation models for networking analysis are more targeted at low voltage use cases [3], [4], [9], [17], [21].

III. Smart Grid Communication Requirements

Among others, the US Department of Energy [15] and the EU Project FINSENY [16] analyzed a range of smart grid applications and identified their different requirements with respect to communication networks (latency, bandwidth, reliability etc.). For instance, they specify that Wide Area Situational Awareness needs latencies below 20ms and bandwidths of more than 500 kb/s. With current narrowband PLC technology over distances encountered in medium voltage grids such numbers are hard to reach. Hence, for applications with requirements in these orders of magnitude, another medium than PLC would need to be chosen. Applications such as Demand Response, Distributed Energy Resources, and grid management with requirements of latencies in the order of 200ms and above and a reliability of 99-99.99% are more realistic for powerline communication. This paper therefore studies latencies and delivery success rates using RPL on MV PLC for such applications.

A typical smart grid application collects information by periodically polling at aggregators (e.g., in substations) to process and issue commands if necessary. Thus most traffic generated by smart grid applications in the medium voltage distribution grid is multi-point-to-point, with occasional point-to-multi-point and point-to-point bursts.

IV. System Overview

The basis of the development and simulation environment of our system is the Contiki OS and its simulator COOJA.

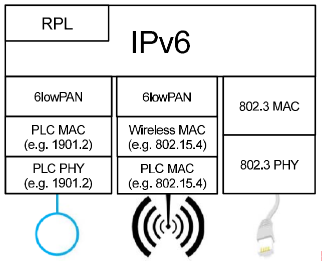

Fig. 1. Example communication stack with three interfaces. Each interface uses a different medium. In our evaluation, we use two PLC IEEE 1901.2 interfaces.

The ContikiOS RPL implementation has been used for a number of studies discussed in Section II. Thus we used it as a starting point for our RPL extensions. The most significant change compared to the original implementation is extending the networking stack for the use of multiple communication interfaces. In our implementation, devices can send and receive messages via PLC and via wireless channels. More precisely there is one common network layer handling several separate (6LowPAN/)link / mac / physical layers. Consequently, RPL had to be generalized for several interfaces in one single networking stack. Having the choice of several interfaces can make communication more robust (another interface can be used if the preferred one fails) and more efficient (the best interface out of all available ones can be chosen separately for each neighboring device).

This enhancement comes at the cost of more control traffic. Since at the network layer neighboring devices can be seen over several interfaces, the same neighboring device is perceived as a separate entity in the network (every interface has a different link local address and a separate neighbor cache storing information on reachable devices). Furthermore, when a node wants to send link-local or global unicast messages to a neighbor reachable via two different interfaces, it should choose the best interface based on the link quality used if the same link local address is in the neighbor cache of two interfaces. To this end uIP had to adapted on the network layer and the objective function had to be extended to compare two links (rather than paths and parents as usual). Our implementation for this issue iterates over all entries the neighbor caches of all interfaces to choose the interface with the best result when applying the objective function on the available link metrics (ties are broken arbitrarily).

While an implementation for the OSI Layer 2 of IEEE 802.15.4 existed already, we had to implement it for the power line communication standard IEEE 1901.2. The main difference can be found in the implementation of the CSMA/CA mechanism. The Contiki implementation for IEEE 802.15.4 uses a simple increasing backoff mechanism. The unslotted CSMA/CA for the mandatory mode of IEEE 1901.2 (normal priortiy packets) keeps track of the contention windows from previous transmissions, i.e., it adapts its behaviour to the current traffic load. In the next section, we explain how the carrier sensing by the physical layer is modeled.

For testing and evaluation, we used COOJA, the Contiki network simulator, developed in Java. An important difference to other network simulators is the fact that it is a system simulator as well: "A simulated Contiki Mote (node) in COOJA is an actual compiled and executing Contiki system. The system is controlled and analyzed by COOJA. This is performed by compiling Contiki for the native platform as a shared library, and loading the library into Java using Java Native Interfaces (JNI). Several different Contiki libraries can be compiled and loaded in the same COOJA simulation, representing different kinds of sensor nodes (heterogeneous networks)" [7]. In addition to the simulations, we verified that our protocol extensions are running on hardware by deploying them on Arduinos communicating over Ethernet and radio.

The next subsection describes our medium voltage power line communication model, which was developed as a component for COOJA.

V. Medium Voltage PLC Model

In order to simulate the behavior of medium voltage power line communication more realistically, we implemented a coarse but simple Signal to Noise plus Interference Ratio (SINR) communication model for COOJA. This model takes concurrent transmissions into account for signal degradation and collisions. In our model, link quality and packet success rate depend on the received signal strength from concurrent senders. This signal strength exhibits a distance-dependency of SNR, which is expected to be the main effect for OSI layer 2 and 3 protocol analysis in regional area MV networks [13], [17]. In contrast, in LV networks electromagnetic interference and impedance effects from electric appliances may dominate, rather than distance [4].

In PLC, all devices that are physically connected by a powerline and are not too far apart can communicate with each other directly in a one-hop manner, although the achievable communication quality may vary. The packet success rate PSR of the MV-PLC model for a single packet thus depends on the distance, as measured by the shortest path through the PLC mesh network connecting the transmitter, the receiver and simultaneous transmissions. For each connection, we compute the SINR, representing the ratio of signal power to the power of interference from concurrent senders plus noise. Only a signal with a high SINR can be received and decoded by a destination. If several senders send a message simultaneously, their signals interfere. For the link from sender node j to receiver node i, the SINR is defined as $SINR_{i,j} = \frac{P_{i,j}}{I_i + N_i} = \frac{P_{i,j}}{\sum_{k \neq j} P_{i,k} + N_i}$, where $P_{i,j}$ represents the received power from sender j at node i, I_i represents the interference power from other currently active senders in the vicinity of node i and N_i the background noise. Assuming identical noise levels and transmission powers at all nodes, this can be rewritten in terms of Signal to Noise ($SNR_{i,j}$) and Interference to Noise ($SNR_{i,k}$) ratios, as $\frac{SNR_{i,j}}{\sum_{k \neq j} SNR_{i,k} + 1}$. From transmission line theory, it is known that signals on a powerline are attenuated exponentially with respect to the distance d. Normalizing signal power by the noise power results therefore in $SNR_{i,j} = SNR_0 \cdot e^{-\gamma \cdot d(i,j)}$, where SNR_0 corresponds to the highest possible SNR (at distance 0), γ is a constant in dB/m which depends on the characteristics of the cable and the PLC operation frequency and $d(i,j)$ is the distance between j and i in m. Note that this simple channel model describes the dominant distance-dependent behavior of the MV PLC channel, as relevant for protocol and routing simulations, but the model would not be sufficient for detailed physical layer studies.

In our extension for COOJA, the SINR is evaluated for each receiver node whenever a sender starts or stops a transmission. If a new sender starts transmitting, the SINR values of all currently active links change to a smaller value. Only a sender-receiver pair with a high enough SINR value for the whole duration of the transmission will have a high probability that the packet is received properly. Otherwise, the transmitted packet will collide with interfering packets. In the simulation of the IEEE 1901.2 CSMA protocol, the channel is considered free if the signals (SINR) from all other currently active senders are below a configurable threshold.

Our MV PLC communication model sets the packet success rate PSR for each transmission, based on the SINR and the packet length (n bits) to be $PSR = (1 - erfc(\sqrt{SINR/2})/2)^n$, where $erfc(z) = 2/\pi \int_z^{-\infty} e^{-t^2} dt$ as suggested in [13]. This assumes Gaussian noise and interference, independent bit errors, and QPSK modulation. Practical PLC systems employ more advanced modulation and coding schemes, but the present simple AWGM model is considered sufficient to model the behavior of PSR for MAC and routing simulations. In the simulation, the successful reception of each packet transmitted is random with the success probability given by the calculated PSR when starting the transmission. For concrete numbers, we base our model on two assumptions [13]: (i) for a transmission without concurrent senders, the packet success rate of a 50 bytes packet is 80% at a distance of 2 km between the sender and the receiver, which corresponds to a bit error rate $BER = 5.6 \cdot 10^{-4}$. (ii) the highest possible transmission success rate is at most $1 - 10^{-6}$, corresponding to $SINR_0 = 15.3$ dB.

In the implementation of the MV PLC model in COOJA, we built a wrapper around the directed graph radio model DGRM. This model offered by COOJA can be used to describe a network by adjusting the success rate for each pair of nodes; however it would be tedious to manually configure a given topology. We provide a simple MV PLC Configurator, where a user can specify a PLC graph and the underlying DGRM graph is computed automatically.

VI. MV PLC Benchmark Scenario

In RPL, nodes periodically broadcast DIO messages to advertise routes towards the root, and send DAO messages to parent nodes to update their routing table. More details are given in the companion paper [2]. We created the following simulation scenario to assess our system for routing in medium voltage PLC networks and to determine whether it is suitable to satisfy the requirements of Smart Grid Applications mentioned in Section III. A simple data collecting application is used, in which the nodes in the network periodically send data messages to one of several sinks (DODAG roots) as soon as they are connected to a DODAG. These messages are routed to the sink nodes through the network, i.e., they are forwarded to the root according to RPL from preferred parent to preferred parent. For the data collection application, each node is assumed to periodically send a status and measurement message. The purpose of the simulations is to determine the message success rate and latencies.

After running the network for 500s to allow the routing protocol to converge to a steady state, the global repair mechanism of RPL is initiated. This enables us to observe the convergence to a DAG that is less dependent on potential startup inefficiencies and to observe the time it takes until all nodes are attached to a valid DODAG again.

A. Network Topology

We evaluate our approach on the topology of a real urban electrical grid of a town in Switzerland (population approx. 20k, area approx. 14 km^2), see Figure 2. It contains 93 nodes (Ring Main Units and Substations), out of which 8 are selected to be sinks; they collect information sent by the non-sink nodes. The edges represent the actual MV power lines between substations. Since the lengths between the stations cannot be inferred from the available topological grid map, we chose them uniformly at random between 200m and 2000m, which is typical for urban environments. The conditions and quality of message reception is described in Section V.

To ensure that Ring Main Units can communicate on the MV power lines attached to them, even if an electrical switch, breaker or transformer between them prevents direct communication among them, as depicted in Figure 3, ideal MV PLC routers are equipped with two couplers. The nodes in our benchmark use two PLC interfaces. Therefore, the whole networking stack has been generalized to support multiple interfaces, as mentioned earlier (the nodes could also be configured to support interfaces of different types such as PLC and radio). Assuming that the two interfaces are connected via the PLC medium increases the number of messages that the nodes have to process.

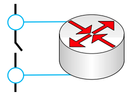

Fig. 3. A PLC router with two couplers.

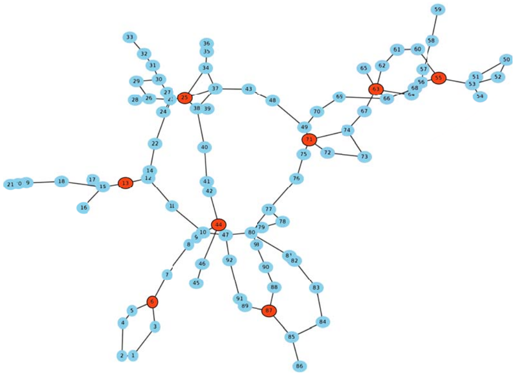

Fig. 2. Benchmark network with 93 nodes and 8 sinks. Sinks (i.e. the DODAG roots in RPL) are represented in red and the regular nodes in blue. For some evaluations a 23 node sub-network was used to reduce the simulation time (after having verified that the differences of salient performance characteristics of the two networks are minimal). Scalability is not an issue because the chosen 23-node subnetwork keeps the characteristics of the 93 node network regarding the sinks/senders ratio.

B. Evaluation Parameters

The goal of the evaluation is to see how much the network and application conditions can be constrained, while still obtaining satisfactory performance. In particular, we are very interested in the message delivery success rate, average and maximum latencies. In addition, the convergence time, the stability of the DODAGs and the overhead required to maintain them are considered as secondary performance criteria. The DODAG stability is measured by the number of changes that occur for the nodes' preferred parents (which are used to establish the routes up towards the root). The overhead is measured by the number of RPL specific control messages such as DIOs and DAOs.

C. Simulation Parameters

The main parameters that were varied during the simulations were the data transmission rate and the message sending frequency. For the data transmission rate, we have chosen representative values for current PLC communication channels, of 100 kb/s and 50 kb/s. These are realistic rates achievable by IEEE 1901.2 links under moderate channel conditions. In practice, data rates may vary due to link quality variations which may lead to bottlenecks. The simulations could easily be extended to model such effects. Data messages were sent once in every T sec interval, with $T = 10s$, $5s$, and to test the system in a limit case, $T = 2s$. The actual transmission instants were selected uniformly at random within each interval T. Hence, the time between two consecutive message transmissions can vary between a minimum of $1ms$ up to $2{\cdot}T$.

The behavior of the network was tested with the objective function provided in the original Contiki implementation, i.e., the minimum rank-hysteresis objective function, described in IETF RFC 6719, using ETX as an additive metric. In the ContikiRPL implementation ETX represents how often the CSMA/MAC layer tries to transmit a packet and if it is successful. Thus, implicitly, ETX includes a measure for congestion. If many neighboring nodes attempt to send a message at the same time the value of ETX will be high, since the packet success rate PSR is low, due to interference according to Section V. ETX is updated each time a packet is sent based on the status of the transmissions and the number of the attempts needed to reach the MAC_TX_OK status. To enable a smooth performance, a first order LP filter is used on the current value $packet\_metric$ received from the MAC layer denoting the number of CSMA re-tries: $ETX_t = 0.9 \cdot ETX_{t-1} + 0.1 \cdot packet\_metric$.

Throughout the simulations, the data packets consist of 64 bytes payloads without headers (representative for messages described by the IEC870-5-104 standard).

VII. Evaluation

As described in the previous section, the main parameters that were varied for this part of the evaluation were the data transmission rates and the data packet transmission interval. We began with a data rate of 100 kb/s and a period of one message every 10 seconds. Then, by lowering the data transmission rate and increasing the frequency of sending packages, the difficulty of the simulation conditions gradually augmented. For every configuration (consisting of a data transmission rate and a message sending interval) we ran 30 simulations, with different random seeds. The seed is used to calculate a random amount of time it will take for each node to complete its boot sequence and begin executing its program. The same random seed is also used by the CSMA protocol, to compute the back-off interval.

A set of 30 random seeds was used to produce 30 simulations of all the different configurations. All of the simulations were run on the topology described in Figure 2. For the time plots, we show averages over sliding windows of 30s, with a step of 5s between two consecutive windows (i.e. the first window covers 0s to 30s, the second one 5s to 35s etc.).

A. Success Rate and Latency

As a first criterion we consider the average end-to-end success rate as a function of the hop count from the root.

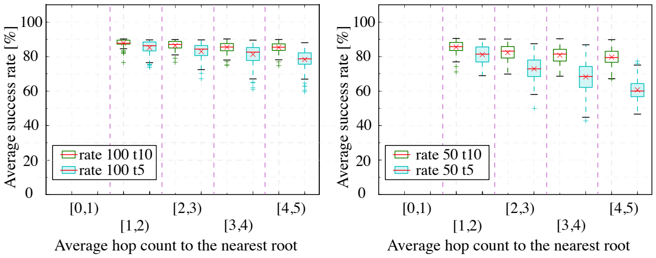

Fig. 4. Box plots of success rate per average hop count to the closest root for transmission rates 100 kb/s and 50 kb/s. As expected sending messages more frequently (cf. $T = 10s$ and $T = 5s$) decreases the average success rate and increases the variance.

In Figure 4 node-bins lie on the x-axis, sorted by the average hop distance to the root. Since a node's position in the DODAG(s) is not fixed during the simulation, the hop count was computed as a weighted average of the distance to the root, where the time the node had the respective hop count value represented the weight of the value. A packet is transmitted successfully over n hops if each of the n transmissions is successful, taking into account the individual packet success probabilities PSR according to Section V. Concurrent transmissions are considered in these computations.

The success rate is shown along the y-axis, in percentages. A box plot was created for each hop count, with a green cross indicating the mean success rate. In this setting, the success rates are very high, regardless of the distance to the roots.

Next we compare average end-to-end (regular-node-to-root) latency values during 30s windows measured per node. In other words, each 30s window contributes one data point for each node (the average latency over all packets received in this window), which are then collected in bins per hop distance and depicted as a box plot. We see that the average latency increases with the node hop count, which is to be expected. With a sending interval of 10s and a transmission rate of 100 kb/s, none of average latencies exceed 200ms. When considering maximum latencies, they are below 400ms at four hops, 280ms at three hops, 180ms at two hops and 150ms at 1 hop. The variance is introduced by random choices depending on the seed value for the start-up time, CSMA back-off interval decisions, PSR-based message delivery etc.

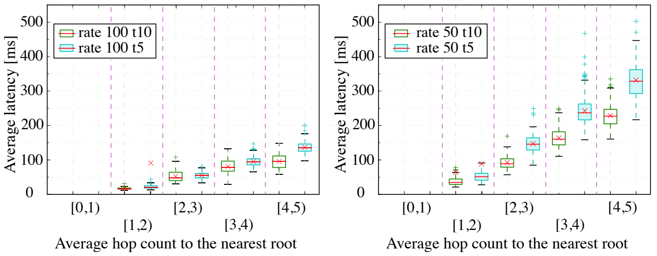

Fig. 5. This figure presents box plot of the average latencies for hop bins with transmission rates 100 kb/s and 50 kb/s with $T = 10s$ and $T = 5s$.

While the average latencies in the cases where messages are sent every 10 seconds are under 200ms, the latencies for the 5s message period are higher than desired. The maximum latencies for nodes at hop distance four and five are below 200ms for rate 100 kb/s and a 10s period and below 350ms for a 5s period. When decreasing the data rate to 50 kb/s the maximum latencies can rise up to 500ms for a 10s period and up to 1s for a 5s period. This, along with the decreased success rates are indicators of the network's level of congestion at 50 kb/s with a 5s period: messages are dropped, and even the ones that reach the destination are significantly delayed if the source is several hops away from the root.

In summary, the network performs in a satisfactory manner for the loads of one message every 10 or 5 seconds and a transmission rate of 100 kb/s. The network becomes slightly congested with a data rate of 50 kb/s and a period of 5s, for a higher frequency of messages the success rate is below 80%. To support a high message sending frequency, one could lower the DIO and DAO transmission rate to lower the control traffic (at the expense of reaction time), use one PLC interface only (less multicast traffic, but can lead to a loss of connectivity) or select a different set of sink nodes.

B. Control Traffic and Stabilization

The next two plots (Figure 6) show the average number of RPL control messages received per node, and the average number of default parent changes for the whole network, over time windows of 30s. We observe peaks in the number of control messages and default route changes when the simulation starts and at 500s, when a global repair is triggered. On average less than 1 node changes the default route per 30s window, after the Trickle algorithm convergence time. Moreover, one can notice that the network stabilizes in less than 120s, producing an average of 60 control messages received per node per window, i.e., 2 messages received per second on average, and hardly any default parent changes. Steady state is regained shortly after the global repair, in less than 100s. The root nodes receive DAOs from all nodes belonging to their DODAG, i.e., communication is possible in both directions. Increasing the message sending frequency does not lead to a change of behaviour: the same interval is needed for convergence and the same number of control messages are sent in a stabilized system. There is change with 50 kb/s compared to the 100 kb/s transmission rate case. When messages are sent once every 5s with 50 kb/s, the system tries to find alternative routes because of the poor success rate. However, the system does not re-stabilize as well, since the whole network is slightly congested. As a consequence, the default routes are changed more often than in other scenarios.

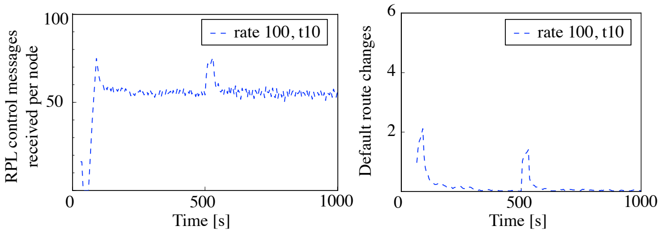

Fig. 6. Average number of received RPL control messages *(left)*, average number of nodes changing default routes *(right)* per 30s window. The nodes start at time 0 with random offsets, at 500s the roots trigger a global repair.

When considering the number of messages sent per node per second with $T = 5s$ during network startup, the average number of data (forwarding) and control traffic are 0.51 and 0.47 messages per node per second. In a stabilized network, the corresponding numbers are 0.55 and 0.35 messages per node per second respectively. If we look at the length of the messages (with IP headers) of 122 bytes for data messages, 104 bytes for DIOs and an average length of 100.8 for DAOs, then the ratio of the total traffic used by the data is 65.3% while the DIO and DAO control traffic use 23.4% and 11.3% respectively. This ratio can be influenced by setting the frequency of DIOs and DAOs trough the trickle timer configuration values. When reducing the control traffic, the network becomes less reactive in the case of failures and topology changes, i.e., a tradeoff between reactivity and network load exists.

VIII. Conclusion

In this article we studied routing on medium voltage PLC, a challenging problem, due to low bandwidth and the effect of packet loss. We extended the Contiki networking stack with the IPv6 routing protocol RPL for PLC to work with multiple interfaces. To evaluate the system performance, we used a simple medium voltage power line channel model and implemented it as a component for the networked system simulator COOJA. Additionally, we constructed a benchmark scenario for smart grid applications using medium voltage PLC and evaluated our system on it. We demonstrated that with a default configuration the system reaches the required performance for light traffic load and favourable data transmission rate. If the load increases and the transmission rate is cut by half, the system struggles to meet the requirements and the configuration might need modification. A companion paper [2] discusses failure recovery in RPL. As a simulation cannot replace a real deployment, these results may not be accurate in practice. Nevertheless, they provide an indication for the suitability of RPL on medium voltage PLC for smart grid applications.

References

[1] E. Ancillotti, R. Bruno, and M. Conti. The role of the rpl routing protocol for smart grid communications. *IEEE Communications Magazine*, 2013.

[2] O. Balmau, D. Dzung, and Y. A. Pignolet. RPL on Speed: Recipes for Faster Failure Recovery. *IEEE SmartGridComm 2014*.

[3] L. Ben Saad, C. Chauvenet, B. Tourancheau, et al. Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: Two Cases Studies. In *SENSORCOMM*, 2011.

[4] G. Bumiller. Power-line Physical Layer Emulator for Protocol Development. In *ISPLC*, 2004.

[5] C. Chauvenet et al. A Communication Stack over PLC for Multi Physical Layer IPv6 Networking. In *Smart Grid Communications*, 2010.

[6] T. Clausen and U. Herberg. Comparative study of rpl-enabled optimized broadcast in wireless sensor networks. In *ISSNIP*, 2010.

[7] F. Osterlind et al. Cross-level sensor network simulation with cooja. In *Local Computer Networks*, 2006.

[8] S. Galli, A. Scaglione, and Z. Wang. Power Line Communications and the Smart Grid. In *IEEE SmartGridComm*, 2010.

[9] E. Malacasa and G. Morabito. Characterization of PLC Communication Channel: a Networking Perspective. In *WSPLC*, 2009.

[10] N. Bressan et al. The deployment of a smart monitoring system using wireless sensor and actuator networks. In *SmartGridComm*, 2010.

[11] N. Tsiftes et al. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. In *CCR*, 2010.

[12] A. Patel, J. Aparicio, N. Tas, M. Loiacono, and J. Rosca. Assessing communications technology options for smart grid applications. In *IEEE SmartGridComm*, 2011.

[13] Y.-A. Pignolet, I. Rinis, D. Dzung, and A. Karaagac. Multi-Interface Extensions for PLC / Wireless Simulator. In *WSPLC, CoRR, http://arxiv.org/abs/1209.6182*, 2012.

[14] N. Saputro, K. Akkaya, and S. Uludag. Survey a survey of routing protocols for smart grid communications. *Comput. Netw.*, 2012.

[15] P. Senior. Key Considerations for Grid Communications Technology. *Electric Light and Power*, 16(1), 2011.

[16] T. Kyntäjä et al. Finseny d2.3 v1.2 distribution network functional architecture description. *FI.ICT-2011-285135*, 2013.

[17] T. Ropitault et al. Realistic model for narrowband plc for advanced metering infrastructure. In *IEEE SmartGridComm*, 2013.

[18] J. Tripathi, J. de Oliveira, and J. Vasseur. A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks. In *CISS*, 2010.

[19] J. Tripathi, J. de Oliveira, and J. Vasseur. Applicability study of rpl with local repair in smart grid substation networks. In *SmartGridComm*, 2010.

[20] N. Tsiftes, J. Eriksson, and A. Dunkels. Low-Power Wireless IPv6 Routing with ContikiRPL. In *IPSN*, 2010.

[21] F. Versolatto and A. Tonello. Analysis of the PLC Channel Statistics using a Bottom-up Random Simulator. In *ISPLC*, 2010.

[22] D. Wang, Z. Tao, J. Zhang, and A. Abouzeid. Rpl based routing for advanced metering infrastructure in smart grid. In *ICC*, 2010.