

Energy Efficiency Impact of Transient Node Failures when using RPL

Ulf Kulau, Silas Müller, Sebastian Schildt, Arthur Martens, Felix Büsching, and Lars Wolf

Institute of Operating Systems and Computer Networks, TU Braunschweig

Email: [kulau|muelle|schildt|martens|buesch|wolf]@ibr.cs.tu-bs.de

Abstract—The Routing Protocol for Low power and Lossy Networks (RPL) is a wide-spread, flexible routing protocol specifically designed for energy-constrained networks such as Wireless Sensor Networks (WSNs). The resilience of RPL has been verified in several existing evaluations, but how a simple node reset can impact the energy efficiency has not been investigated so far. Thus, in this paper we examine the energy costs of transient node failures in a RPL network. We show that such temporary node failures can have a significant impact on the network's total energy usage. As transient node failures are common in WSNs this work reveals an important weakness of RPL which can even be exploited for attacks.

I. INTRODUCTION

Due to energy constraints common WSN nodes have limited processing capabilities and use low-power, low-bandwidth RF communication. WSNs are usually designed to be resilient as a complete network. However, individual nodes can fail for a number of reasons, e.g., due to hard- or software errors, attacks, lack of energy or being isolated from the network due to the RF environment. Combined with the fact that WSNs are often deployed in adverse environmental conditions a single node can be quite unreliable. To allow multi-hop data forwarding in such low-power and lossy networks, many routing protocols have been proposed [1], [2]. In a standardization effort the IETF Routing Over Low power Lossy networks (ROLL) working group designed the dedicated RPL protocol [3] for such networks. RPL itself is adaptable to application specific routing metrics and goals. As RPL can run on top of the common 6LoWPAN [4] networking stack it is quickly becoming the de-facto routing standard in many WSN implementations.

Under conditions of high network churn, i.e., when nodes fail and vanish from the network just to reappear a short while later and must be re-integrated into the network, the restructuring operations performed by RPL potentially cause a significant overhead in terms of energy consumption. This can especially be a problem if a node is only intermittently available causing frequent RPL updates. Such a scenario is not uncommon and may happen due to a cyclic software error, energy shortages, hardware problems or an active attacker. Until now, this additional power consumption of RPL due to transient node failures has not been investigated.

In this paper, we present an extensive evaluation analyzing the negative impact of intermittently unavailable nodes in RPL. It will be shown that even a single node reset can impact the energy efficiency of the entire network. Depending

on the topology and location these impacts of node resets are not locally bounded but can also drain energy from surrounding nodes. Except from adverse effects of common WSN reliability problems, the results show that induced node resets by an attacker can be an efficient attack method, shorting the lifetime of a network.

II. RPL OVERVIEW

In RPL, for each sink node a dedicated Destination Oriented Directed Acyclic Graph (DODAG) is constructed which describes the routing paths. Thus, when a path to a specific destination is established in RPL a node can always forward information if it has directed, outgoing links for that destination. To distribute information about the state of a RPL network, each node sends DODAG Information Object (DIO) messages scheduled by a trickle timer. Among other information, a DIO includes the rank of a node which is used by other nodes to select the best next hop on the path to the sink. A next hop node that is chosen for a specific destination is called the *preferred parent* as it is one of the parent nodes in the DODAG for that destination. A preferred parent is selected by using an objective function, that enables RPL to adapt to different optimization criteria depending on the chosen objective function. DODAG Information Solicitation (DIS) messages are used by nodes to request such DIOs from their neighbors. To allow downward-routes from the sink to other nodes, Destination Advertisement Objects (DAOs) are sent from child to parent nodes.

When a node fails, and thus a path in the DODAG becomes unavailable, RPL supports *local repair* and *global repair* mechanisms to restore the DODAG to a state where every node is again able to choose a preferred parent to forward data to the destination. Therefore, RPL is designed to deal with topology changes that can occur when nodes fail. Overall, RPL is a dynamic routing protocol and generally well suited for the requirements of WSNs.

III. RELATED WORK

Several studies looked into the performance of RPL such as [5], [6], [7] and a comprehensive evaluation of routing performance metrics was performed in [8]. The OMNeT++ network simulator was used to look into end-to-end delays, expected transmission count (ETX) also considering RPL specific configuration parameters such as timer settings for local- and global-repair. However, while many of these evaluations focused on typical Quality of Service (QoS) metrics in various WSN topologies, no systematic evaluation exists which studies

how the energy consumption of WSNs using RPL is affected under adverse conditions such as unreliable nodes.

The control overhead of RPL is investigated in [9]. The authors observe that during the bootstrapping process of a WSN the overhead is comparatively high due to the construction of the DODAGs.

The effectiveness of the RPL repair process has been studied in [10]. The Contiki RPL implementation and the Cooja simulation environment have been used for the evaluations. They show that RPL's repair process is dependable, but of limited reactivity. Thus, RPL is not well suited for mobile nodes with frequent topology changes [11].

A rather critical evaluation of RPL has been performed in [12] where several possible weaknesses and limits of RPL such as suboptimal handling of loops were revealed.

While [8] and [10] indicate that permanently disappearing nodes may be a problem, up to now the impact of temporary node failures has never been analyzed in detail. Compared to mobile nodes, which also behave like as appearing and disappearing nodes, temporary failing nodes will lose their RPL state. In such a scenario the self-healing character of RPL guarantees reliable routing [10]. But in the following we will show that the aftermath of even one malicious node affects the energy consumption significantly.

IV. TRANSIENT NODE-LOSSES IN WSNs

Due to their design and deployment many WSNs struggle with rather unreliable nodes. Low-grade hardware, energy constraints as well as rough environmental conditions could cause various errors. Therefore, nodes are often equipped with a watchdog that resets the entire node to recover from a failure state. In this work we mainly consider this type of error where unexpected resets/reboots of nodes lead to transient node losses in a WSN. Node resets can occur if nodes are equipped with an unreliable energy source. To allow long lifetimes of WSN applications, many nodes are supplied by energy harvesters [13]. Frequent, unpredictable and transient losses might occur when nodes run out of energy. In [14], [15] the voltage level of sensor nodes is reduced below the recommended specifications to achieve substantial energy efficiency gain. Although the intention is a dependable system even when undervolting, absolute reliability can not be guaranteed.

Another challenge in WSNs which are deployed outdoors are harsh environmental conditions. Especially the temperature could affect the link quality [16] but also the proper functioning of the entire node. The authors in [17] reported that nodes stopped working under extreme temperatures and had to be reset by external watchdogs. With regard to long-term WSN deployments and accelerated by extreme temperatures, the aging of components diminishes the reliability of nodes, too [18], [19].

Overall, transient node failure in WSNs is the rule rather than the exception. The knowledge that resetting nodes may harm the entire WSN may motivate the use of intentionally, induced node resets as possible attack vector. Indeed, existing work deals with potential attacks on WSNs [20], [21], however forced node resets have not been considered yet.

TABLE I: CONTIKI RPL TEST CONFIGURATION. PER DEFAULT CONTIKI-MAC [27] WAS USED FOR RADIO DUTY CYCLING.

Parameter	Value	Comment
RPL Mode of Operation	Storing/Multicast	Contiki Standard
RPL Objective Function	ETX [28]	Contiki Standard
RPL DIO_interval_min	12	Contiki-RPL Standard
RPL DIO_interval_doublings	8	Contiki-RPL Standard

Such node losses may cause the routing protocol to react when losing the node and when reintegrating it into the network. Usually, a rebooting node will lose its routing state, as it is often kept in RAM for efficiency reasons. In this work we analyze the effects of such transient node failures with a loss of state on RPL.

V. TEST SETUP

Our experiments were done using the Cooja simulation environment [22]. The CoojaTrace plugin [23] was used to profile the power consumption of the simulated nodes. The energy consumption estimate of CoojaTrace is based on the accumulation of times a component of a node stays at a discrete power state such as an active or sleeping CPU, or a transmitting or receiving radio transceiver.

All simulations are based on standard TmoteSky [24] nodes running the RPL implementation of Contiki [25] with the default configuration as shown in Table I. The default values of Contiki RPL do not conform to the default values recommended by the IETF Standard [3], however [26] has shown that the Contiki defaults achieve a good routing performance.

The period of a single simulation run was set to $t_{run} = 10$ min. At a random point in time $2 \text{ min} \leq t_{rst} \leq 9.5 \text{ min}$ during a simulation run, restarts were triggered. This gives the network time to bootstrap the RPL DODAGs and stabilize before we restart nodes and observe the impact. We found that the exact time of the reset during the chosen period does not change the impact of the reset.

Each experiment always includes 1000 simulation runs and was compared against a reference simulation of the respective scenario based on 2000 simulation runs without any resetting nodes.

Every 5s each node generates a packet with a payload of 24B and sends it to the sink. According to [26] this data traffic pattern is optimal for the default parameters of the Contiki RPL implementation. To avoid impacts from potentially unreliable links between nodes, the link quality of all links was set to 100% in Cooja.

VI. BASIC SCENARIO

To get a basic understanding of the impact resetting nodes have on a WSN, we simulated a simple topology consisting of 8 nodes as shown Figure 1a. In this scenario node ID1 is the sink node.

A. Single Node Reset

In this experiment we force a single node to reset within the timeframe (t_{rst}) described in Section V. After being reset the node immediately restarts and joins the network again.

When comparing the energy consumption we do not include the sink node, as naturally in this scenario the sink node

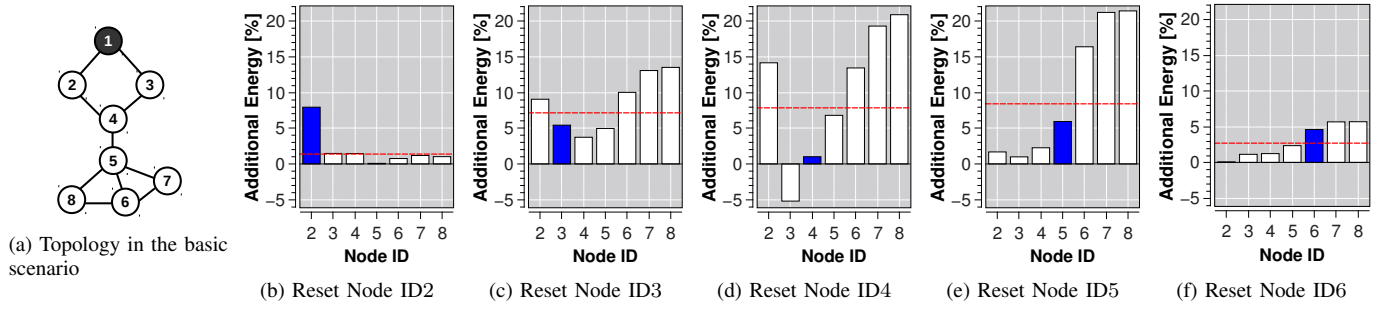


Fig. 1: Energy consumption of nodes when a single node resets (red line indicates additional needs for entire network).

is the most loaded node with the highest duty-cycle. Its energy consumption reaches up to 80 % of the total network energy consumption. However, for many deployments it is a reasonable assumption that the sink node is mains-powered. Thus, energy efficiency is most critical for the other nodes operating on a battery or relying on energy-harvesters.

Figure 1 summarizes the results of resetting different nodes. Only one node was reset once in each run. The restarted node is indicated by the blue bar in the figures. We show the relative increase in energy consumption for each node as well as the entire network (as indicated by the red line in the graphs) compared to a reference run with no node resets. The Figures 1b and 1c show the effect of resetting either node ID2 or node ID3. Although the nodes seem to be equal from a topology point of view, as they act as redundant links to the sink node, there is a significant difference in the additional energy consumption when one of the nodes reset. While a reset of node ID2 increases the overall energy by only 1.3 %, the reset of node ID3 leads to an additional energy consumption of 7 %. The reason is that in most simulation runs node ID4 selects node ID3 as its preferred parent. Thus, a reset of node ID3 requires repairing the DODAGs containing node ID3. Resetting a node that plays as more prominent role in routing data is more expensive in terms of energy consumption.

The Figures 1d and 1e show the additional energy consumption when resetting node ID4 and ID5, respectively. These two nodes are bottlenecks in this topology. A fault of node ID4 or node ID5 leads to network partitioning. The nodes ID{6, 7, 8} will lose their connection to the sink temporarily and have to reestablish workable routes once node ID4 or ID5 are available again. In such cases the average energy consumption of the WSN is increased significantly by about 8 %.

Nodes ID{6, 7, 8} do not relay data from other nodes. They just transmit their own data to the sink periodically. Figure 1f shows the result when node ID6 is forced to restart. Although node ID6 is neither a part of the routing path of node ID7, nor of node ID8, a reset of node ID6 still affects the energy consumption of these nodes. As a reset leads to a loss of node ID6's routing state, node ID6 has to complete its RPL bootstrapping again. This means the exchange of several RPL control messages which results in a higher energy consumption not only for node ID6, but also for neighboring nodes ID7 and ID8. However, compared to the loss of nodes crucial to already used routing paths, the total energy consumption increase of 2.8 %, is less pronounced. Thus, the results from resetting node ID7 or ID8 are similar to node ID6 (and hence omitted here).

The increase in energy consumption is due to RPL overhead adapting to vanishing and reappearing nodes. Table II summarizes the increase of different RPL messages compared to the reference simulation where no reset occurred. Additionally, the table shows the achieved PRR. As expected, the PRR decreases when restarting nodes lead to a temporary network partition or when resetting a node that is a preferred parent on a route used by other nodes.

The results from the simple topology in Figure 1a show that resetting a node can have a significant impact on the energy needs in a RPL network. The severity depends on the role and position of a node that is restarting. Resetting a node also influences PRR, either directly because a missing node partitions the network, or indirectly because some messages get lost while RPL reacts to a lost parent. These results also show that resetting nodes is a suitable attack vector for WSNs. Resetting a node has not only local effects, but can negatively effect the network as a whole. The impact of such an attack can be maximized by selecting a node whose reset will impact the network most. Therefore, in the following section we will look at two typical kinds of WSN topologies and analyze the impact of resetting different nodes in those topologies.

VII. MESH AND BINARY-TREE SCENARIO

The previous section demonstrates that the reset of any node has always an impact on its neighbors. Further, a large influence occurs if the resetting node provides the only connection between subnetworks. To get a clearer picture on those effects under different conditions, we will compare two topologies commonly found in WSN deployments.

The first topology is a binary tree topology. Every node has exactly one path to the sink. Therefore, the graph is only 1-connected, and the removal of any node leads to a partition of the graph. Thus, every non-leaf node in the topology is crucial for relaying data for 1 or more other nodes. The binary tree topology is composed of 31 nodes where node ID1 is the sink. An illustration of this topology is shown in Figure 2a.

TABLE II: SUMMARY OF THE AVERAGE CONTROL OVERHEAD AND DECREASED PRR DUE TO A SINGLE NODE RESET.

		Resetted node						
	<i>Ref</i>	2	3	4	5	6	7	8
DIO (97 B)	<i>56</i>	68	71	73	83	73	73	73
DIS (64 B)	<i>3</i>	4	4	4	4	4	4	4
DAO (76 B)	<i>120</i>	138	148	185	185	165	165	165
PRR [%]	<i>99.2</i>	99.0	96.8	95.0	98.5	99.0	99.0	99.0

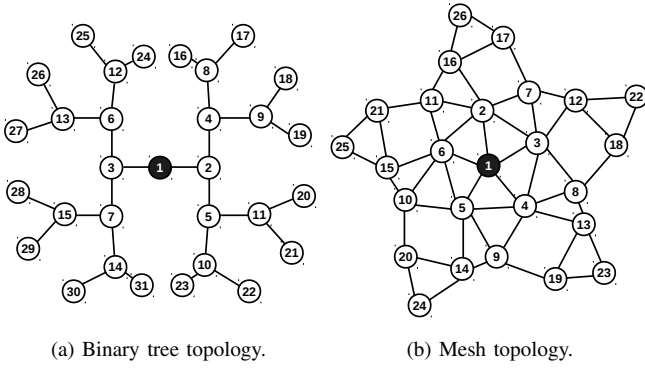


Fig. 2: Topologies used for the in-depth examination.

The second topology to be analyzed is a mesh topology. Each node in this topology has at least two possible parents to choose from when sending data to the sink node and, therefore, several neighboring nodes. The graph is 2-connected and the removal of any node will not lead to a partition of the graph. This topology is shown in Figure 2b and is composed of 26 nodes where the sink node is also node ID1.

In addition to a single reset we will also observe what happens when resetting a node multiple times, which is a likely assumption if, e.g., an attacker causes the reset. Such failure pattern might also be observed if a node resets due to external conditions such as high temperature or lack of energy.

The test setup and parameters are identical to those described in Section V.

A. Single Node Reset

Similar to the basic scenario in Section VI, this evaluation shows the impact of resetting a single node once during a simulation run.

1) *Binary-Tree Topology*: The nearer a node is to the sink in this topology, the more packets it forwards, and the more nodes rely on it for forwarding data. Hence, in this topology the resetting of a node which is closer to the sink, i.e. which connects a larger subnet, should impact the network more. Due to the non-redundant nature of this topology a resetting node means a temporary partition of the network and leads to a decrease of the PRR.

Figure 3 summarizes the results when a node with a distance of 1, 2 and 3 hops to the sink resets. Exemplary for nodes with 1 hop distance to the sink, Figure 3a shows the change in energy consumption of all nodes when node ID2 resets. The subnet of node ID2 contains the nodes $ID\{4, 5, 8, \dots, 11, 16, \dots, 23\}$. It can be seen that node ID2 as well as its subnet suffer from the reset. The distance from the resetting node to a node in its subnet is also of importance. While the energy consumption of node $ID\{4, 5\}$ increases by about 3 %, the energy consumption of $ID\{16, \dots, 23\}$ is about 10 % higher compared to the network without a reset event. This effect is due to the fact that nodes located at the edge of this topology have lower duty cycles as they forward less data compared to nodes located nearby the sink. Thus, the additional overhead due to the node reset is comparatively larger as each node needs to reestablish routes to the sink. Overall, the total additional energy consumption is about 3 %.

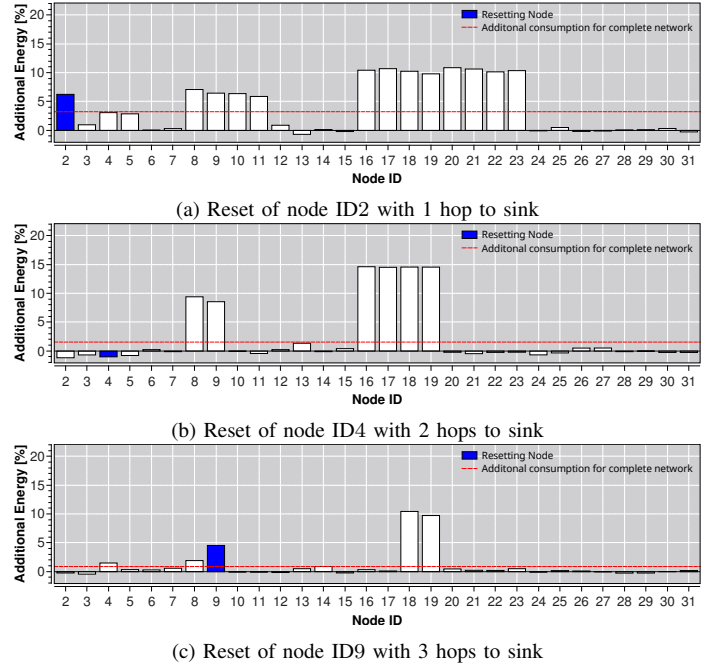


Fig. 3: Energy consumption changes in binary tree topology with single nodes resets.

Considering that the reset of ID2 only affects the right side of the binary tree topology, the energy consumption of the affected part of the network is actually 6 %.

The effect, that mostly the child network of a resetting node is affected, can be observed for failing nodes located 2 and 3 hops away from the sink. For 2 hops distance Figure 3b shows the results of resetting node ID4. As an example for 3 hops distance Figure 3c shows the results of resetting node ID9. Overall, the adverse effects of a node reset declines when the resetting node only forwards data for a small subnet.

Table III summarizes the RPL control overhead and PRR for the complete network. As the considered topology does not offer any redundant links to the sink, a decrease in PRR is inevitable.

2) *Mesh Topology*: Similar to the binary-tree scenario we evaluate the effects of resetting nodes with 1, 2 or 3 hops distance to the sink in the mesh topology.

Figure 4 shows the impact of resetting node ID4 (1 hop to sink), ID16 (2 hops to sink) and node ID25 (3-hops distance). Compared to the binary tree topology the increase of the total energy consumption is slightly lower. In the mesh topology some nodes might already been using alternative parts when a node reset occurs, and affected nodes can change their routes to avoid the missing node. Similar to the observation in the basic scenario in Section VI, it can be seen that the

TABLE III: SUMMARY OF THE AVERAGE RPL CONTROL OVERHEAD AND PRR IN BINARY TREE TOPOLOGY WITH SINGLE NODE RESETS.

	Ref	Reset node		
		ID2 (1 hop)	ID4 (2 hops)	ID9 (3 hops)
DIO (97 B)	217	270	268	237
DIS (64 B)	17	18	18	18
DAO (76 B)	575	850	775	630
PRR [%]	91.0	88.5	89.0	90.5

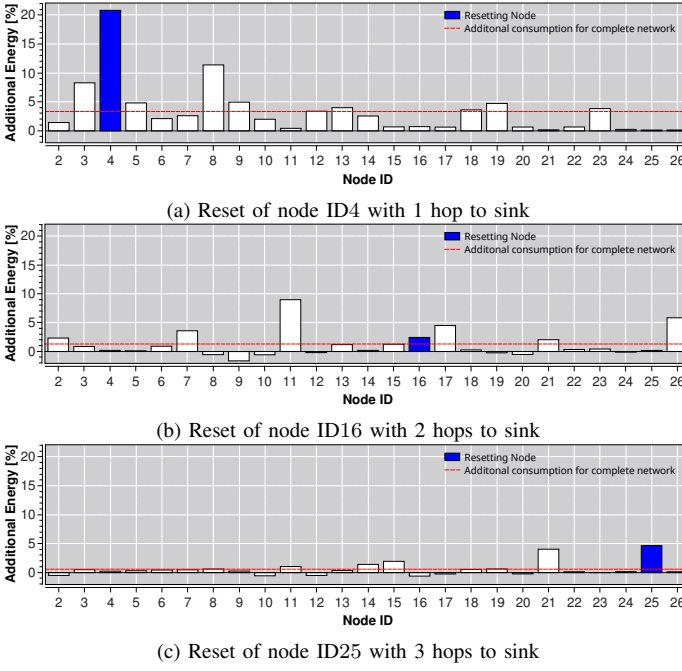


Fig. 4: Energy consumption changes in mesh topology with single nodes resets.

neighborhood of a resetting node is affected most.

It is noticeable that the reset of node ID4 has a significant impact on this node itself (cf. Figure 4a). The additional energy consumption reaches about 20 %. Analysis showed, that this massive effect can not be traced back to the increased control overhead. In fact in some cases the resetting node (in this case ID4) selects a different preferred parent after reboot. In particular nodes ID3 or ID5 might be selected instead of the sink node. Due to the used default Contiki MAC such a setup leads to a much higher radio duty cycle for node ID4 leading to this over-proportional energy consumption increase. While this shows some potential for cross-layer optimization when using RPL in Contiki, this is out of scope for this work. However, despite potential for optimization, in a specific deployment knowing such cross-layer dependencies can also make attacks more effective.

The results of a reset of node ID16 (2 hops to sink) and ID25 (3 hops to sink) are shown in Figure 4b and Figure 4c, respectively. For these test cases the impact on the total energy consumption of the WSN is rather small ($< 1\%$). However, a reset still affects the neighboring nodes.

Table IV summarizes the average control overhead and PRR for the mesh topology. Compared to the binary tree, the mesh topology offers alternative routes when resets occur. Thus, the PRR is less affected and still remains above 99 %. While

TABLE IV: SUMMARY OF THE AVERAGE RPL CONTROL OVERHEAD AND PRR IN MESH TOPOLOGY WITH SINGLE NODE RESETS.

	Ref	Reset node		
		ID2 (1 hop)	ID16 (2 hops)	ID25 (3 hops)
DIO(97 B)	185	218	205	199
DIS(64 B)	13	14	14	14
DAO(76 B)	375	477	415	395
PRR [%]	99.5	99.3	99.3	99.5

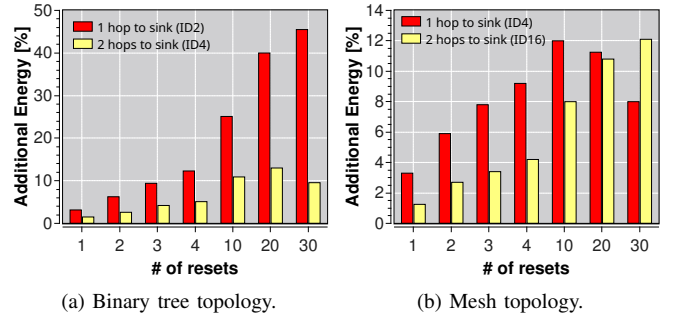


Fig. 5: Energy consumption changes in binary tree and mesh topology with multiple node resets.

a more connected topology offers more robustness against transient node failures, it is still possible to drain energy from nodes by selectively causing nodes to restart.

B. Multiple Node Reset

So far we only considered a singular node reset during a simulation run. The question is, what happens if a node is intermittently available and resets multiple times in quick succession.

Multiple node resets within a short timeframe are not uncommon in real WSNs. The reset reasons described in Section IV might occur periodically. For example in [15] it was observed that an insufficient voltage level might lead to a burst of resets. Besides malfunctions and environmental conditions, an attacker will also choose to reset a node frequently if this increases the effect of the attack. Therefore, we repeated the evaluation in Section VII-A, but forced a node to reset multiple times during a simulation run t_{run} .

1) *Binary-Tree*: The general effects that are caused by the resets are the same as described in Section VII-A1. Thus, here we are only showing the average increase in energy consumption for the complete network. Figure 5a shows the results for multiple resets of node ID2 (1 hop) and for multiple resets of node ID4 (2 hops). It can be seen that every additional reset increases the overall energy consumption additively. While a single node reset of ID2 causes only about 3 % additional energy consumption, 4 resets of this node during one simulation run leads to 12 % higher energy consumption. With more frequent resets the additional energy consumption is even higher. From about 25 % for 10 resets it rises to 45 % for 30 resets. Nevertheless, the higher the reset frequency, the more the additional energy consumption declines as can be seen in Figure 5a where the negative impact of 30 resets of node ID4 is even lower compared to 20 resets of this node which is in line with the added RPL overhead

2) *Mesh*: Similar results can be observed when multiple resets of a node occur in the mesh topology. Figure 5b shows the results for multiple resets of node ID4 (1 hop) and node ID16 (2 hops). As in the binary tree case every additional reset increases the energy consumption of the entire WSN. However, compared to the binary tree topology, the increase in additional energy consumption declines faster with every additional reset. Mainly the objective function ETX used by Contiki is responsible for this effect. ETX estimates how many (re-)transmissions are needed to forward a packet to the sink.

In RPL the goal of the objective function using ETX is to minimize the overall estimated amount of transmissions. The ETX value of a resetting node will increase when that node resets multiple times. When the ETX exceeds a predefined threshold, the surrounding nodes will select another preferred parent. Thus, in a mesh topology where alternative paths exist RPL can slightly mitigate the effects of very unreliable nodes. The utilization of the resetting node decreases and, thus, the impact of continuously resetting the node decreases. For an attacker in a mesh networks this means, in order to generate the most effect, he should reset different nodes following RPL's rebalancing, or, if only one node can be restarted, reduce the frequency of resets to prevent ETX from choosing alternative nodes.

VIII. SUMMARY AND CONCLUSIONS

Considering that unreliable individual nodes are to be expected in many WSN application scenarios, it is important how the routing protocol reacts to it. While the RPL protocol is able to dynamically adapt to changes in topology such as disappearing and reappearing nodes, so far the energy efficiency impact of restarting nodes has not been considered. By performing a comprehensive evaluation, this work revealed that under the effect of restarting nodes in a RPL network, energy consumption might increase significantly, not only for the restarted node, but also for the network as a whole. Understanding these effects and the time constants involved make resetting nodes even an effective attack vector, as the resets will not only directly affect the operation of a node, but can be used to drain lifetime of the whole network.

Using the example of different WSN topologies, we investigated the causes and effects for the increased energy consumption. Depending on the scenario and topology, one malicious node can increase the total energy consumption by up to 8% while individual nodes may consume 20% more. Furthermore, resetting nodes multiple times can amplify this effect.

Such transient node failures may occur accidentally or be caused by an active attack. In future work we will look into ways to harden RPL against conditions of intermittent node failures. Some counter-measures might be of technical nature, such as optimizing standard RPL parameters or making certain state information more persistent and reconcile a saved state in an efficiently when rejoining the network. However, in the case of an active adversary some distributed misbehavior detection might be required at the RPL level to identify an attacker, or at least notify operators about a possible ongoing attack.

ACKNOWLEDGMENTS

This research was partially supported by the German Research Council (DFG) under the grant no. BU 3282/2-1.

REFERENCES

- [1] Al-Karaki *et al.*, "Routing techniques in wireless sensor networks: a survey," *IEEE wireless communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [2] N. A. Pantazis *et al.*, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.
- [3] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), IETF, Mar. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6550.txt>
- [4] N. Kushalnagar *et al.*, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," Tech. Rep., 2007.
- [5] J. Ko *et al.*, "Evaluating the Performance of RPL and 6LoWPAN in TinyOS," in *Workshop on Extending the Internet to Low Power and Lossy Networks (IP+ SN)*, vol. 80. Citeseer, 2011, pp. 85–90.
- [6] T. Zhang *et al.*, "Evaluating and Analyzing the Performance of RPL in Contiki," in *Proceedings of the first international workshop on Mobile sensing, computing and communication*. ACM, 2014, pp. 19–24.
- [7] J. Ko *et al.*, "ContikiRPL and TinyRPL: Happy together," in *Workshop on Extending the Internet to Low Power and Lossy Networks (IP+ SN)*, 2011.
- [8] J. Tripathi *et al.*, "Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6687 (Informational), IETF, Oct. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6687.txt>
- [9] N. Accettura *et al.*, "Performance analysis of the RPL routing protocol," in *IEEE International Conference on Mechatronics (ICM)*. IEEE, 2011.
- [10] K. D. Korte *et al.*, "A study of the RPL repair process using ContikiRPL," in *IFIP International Conference on Autonomous Infrastructure, Management and Security*. Springer, 2012, pp. 50–61.
- [11] K. C. Lee *et al.*, "RPL under mobility," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2012.
- [12] T. Clausen *et al.*, "A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL)," in *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2011.
- [13] S. Sudevalayam *et al.*, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, 2011.
- [14] U. Kulau *et al.*, "Undervolting in WSNs: Theory and Practice," *Internet of Things Journal, IEEE*, vol. 2, no. 3, pp. 190–198, June 2015.
- [15] —, "Idealvolting: Reliable undervolting on wireless sensor nodes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 2, p. 11, 2016.
- [16] C. A. Boano *et al.*, "The impact of temperature on outdoor industrial sensor network applications," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, 2010.
- [17] M. Mousa *et al.*, "Water level estimation in urban ultrasonic/passive infrared flash flood sensor networks using supervised learning," in *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press, 2014.
- [18] A. H. Dehwhah *et al.*, "Lessons learned on solar powered wireless sensor network deployments in urban, desert environments," *Ad Hoc Networks*, vol. 28, 2015.
- [19] F. Reynolds, "Thermally accelerated aging of semiconductor components," *Proceedings of the IEEE*, vol. 62, no. 2, Feb. 1974.
- [20] C. Karlof *et al.*, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, 2003.
- [21] L. Wallgren *et al.*, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks*, 2013.
- [22] F. Osterlind *et al.*, "Cross-level sensor network simulation with cooja," in *Proceedings of the 31st IEEE Conference on Local Computer Networks*. IEEE, 2006.
- [23] M. Strübe *et al.*, "Demo Abstract: CoojaTrace, Extensive Profiling for WSNs," in *Poster and Demo Proc. of the 9th European Conf. on Wireless Sensor Networks (EWSN 2012)*. Citeseer, 2012.
- [24] Moteiv Corporation, "Datasheet: Tmote Sky," 2006. [Online]. Available: http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf
- [25] A. Dunkels *et al.*, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004. IEEE, 2004.
- [26] H. Ali, "A performance evaluation of rpl in contiki," 2012, master's Thesis. [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:833572/FULLTEXT01.pdf>
- [27] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.
- [28] O. Gnawali *et al.*, "The Minimum Rank with Hysteresis Objective Function," RFC 6719 (Proposed Standard), IETF, Sep. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6719.txt>