Transactions on Machine
Learning and Data Mining
Vol. 13, No. 1 (2020) 27 - 39
© 2020, ibai-publishing,
P-ISSN: 1865-6781, E-ISSN 2509-9337
ISBN: 978-3-942952-73-6

SL-RPL: Stability-Aware Load Balancing for RPL

Feng Wang[1], Eduard Babulak[2] and Yongning Tang[3]

[1] School of Engineering, Liberty University, USA
[2] School of Business, Liberty University, USA
[3] School of Information Technology, Illinois State University, USA
ebabulak@liberty.edu

Abstract. The Internet of Things (IoT) is exponentially growing and requires reliable and efficient protocols to grow with it. Routing Protocol for LLN (RPL) is designed to meet the different requirements of IoT. However, RPL does not support load balancing. Without a proper load balancing mechanism, IoT devices may experience severe congestion problem, resulting in packet loss and delay. However, when using congestion-aware routing metrics to achieve load balancing, frequent parent switching may occur, which can induce instability in IoT networks. In this project, we propose a stability-aware load balancing method for RPL. We evaluate the improved performance of our solution in simulations. Our simulation results indicate that the proposed method can significantly improve the stability and performance of RPL.

Keywords: Load Balancing, RPL, IoT, Objective Function

1 Introduction

Routing Protocol for LLN (RPL) is proposed to achieve reliable and energy efficient routing for Low power and Lossy Networks (LLNs). However, PRL does not support load balancing. Without a proper load balancing mechanism, IoT devices may experience severe congestion problem. Because most IoT devices typically have tight resource constraints, such as limited battery, they may quickly exhaust their energy, causing the entire network to disconnect, and resulting in packet loss and delay. Therefore, load balancing is a critical function for RPL. To address this problem, almost all of previous works focus on designing new metrics to select parents in a balanced way. For example, previous work [1] combines Hop count with Expected Transmission Count (ETX) and remaining power parameters. A QoS-aware method based on fuzzy parameters is also proposed in work [2]. A queue utilization and hop distance are used to select parents in [3]. In [4], a load balancing method is proposed by estimating the amount of traffic on a node.

Although many load balancing methods have been proposed, it is still difficult to utilize the proposed methods to achieve load balancing, especially for heterogeneous IoT networks. Specifically, when using congestion-aware routing metrics to implement load balancing, *frequent parent switching*, or *parent oscillation*, may occur. In this paper, we investigate the extent to which the frequent parent changes being observed when different load balancing methods are used. We find that the standard RPL and some existing load balancing solutions can cause a large number of parent changes. We also investigate the impact of frequent parent changes on the performance of IoT networks. We find that IoT networks could suffer from significant packet loss due to frequent parent changes. The problem becomes more severe when heterogeneous nodes generate high traffic and dynamic load.

To migrate frequent parent changes, in this paper, we propose a stability-aware load balancing for RPL, SL-RPL, which can achieve load balancing among nodes while preserving network stability. We have integrated the proposed mechanism with RPL protocol in Contiki OS. Our simulation results indicate that SL-RPL can significantly reduce the occurrence of frequent parent changes, and improve the performance of IoT networks in terms of packet losses and energy consumption.

The rest of paper is organized as follows: We briefly introduce RPL and related works in Section 2. We present the frequent parent change problem in Section 3, and SL-RPL in Section 4. We evaluate the performance of SL-RPL in Section 5. We conclude the paper in Section 6 with a summary.

2 Background and Previous Works

2.1 Overview of RPL

RPL is a distance vector routing protocol specifically designed for 6LoWPAN. The objective of the protocol is to construct and maintain a Destination-Oriented Directed Acyclic Graph (DODAG). The DODAG roots at a border router. RPL uses four types of control messages to detect the neighboring routers, maintain DODAG, and advertise RPL information. The RPL messages are: DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS) and DAOACK.

A root starts the DODAG construction by periodically broadcasting DIO messages. Multicasted DIO messages are periodically sent by nodes for DODAG construction. A DIO message contains general information such as the RPLInstanceID, rank, DODAGID, etc. When a node decides to join the DODAG, it saves a list of potential parents, and selects one of them as its preferred parent. After that, the node generates a new DIO message and broadcasts it. In RPL, a node may wait to receive DIO messages so that it will not send any multicast DIO messages until it has joined a DODAG. Or, a node may decide to send one or more unicast DIS messages to request the DODAG related information from neighboring nodes. According to RFC6550 [5], when a node receives a unicast DIS message, it must unicast a DIO including a DODAG Configuration option to the sender in response.

2.2 Objective Functions

In RPL, an Objective Function (OF) is used to define rules and policies for selecting preferred parents. More specifically, the OF specifies how to compute the rank of a node based on one or a set of routing metrics, such as hop count or link quality. Each node's rank indicates its routing distance to the DODAG root. Based on the rank, the OF specifies how to select the preferred parents. In order to avoid routing loops, the rank must monotonically increase from a root towards the other nodes. Currently, the OF zero (OF0) [6] and the Minimum Rank with Hysteresis OF (MRHOF) [7] are defined in RPL. When using OF0 as the objective function, a rank is calculated by using hop count. For a parent selection, OF0 always selects the parent with least rank as its preferred parent. When using MRHOF [7] as the objective function, a rank is calculated by using ETX, which indicates the link quality between two nodes. The metric is advertised in the Metric Container of DIO messages. When using MRHOF, a node selects the parent with the least number of expected transmissions as its preferred parent. Unlike OF0, MRHOF attempts to address the frequent parent change issue by using a hysteresis mechanism [7]. More specifically, a node switches to a new parent only if the new minimum calculated path cost is smaller than the preferred parent's path cost by a pre-defined threshold. Note that the RPL standard [5] does not mandate any particular OF nor routing metric to be used.

RPL initially does not consider load balancing so that the routing metrics, ETX and shortest distance (hops) are less efficient in handling heavy traffic than light traffic. The consequence is that a network will experience severe congestion problem, which results in packet loss and delay. Previous work has shown that the majority of the packets in RPL are lost due to the buffer overflow [8]. The problem becomes more critical when the child node does not change their current parent and keep sending data traffic even though its parent node that is already overloaded.

2.3 Previous Works

Many solutions have been proposed to address the load-balancing problem in RPL. We briefly present those methods according to their routing metrics and notification methods.

Routing Metrics. *Many metrics are proposed to indicate the degree of congestion.* The first metric is queue utilization [8,3]. This metric reflects the buffer space in a potential parent node. For example, QU-RPL [3] defines a buffer occupancy ratio of a node to indicate the number of packets in the queue. The second metric is Packet Delivery Ratio (PDR) [8,4,9,10]. For example, in [8], the number of packets being forwarded by the candidate parents is used. In [4, 9], a load balancing method is proposed by estimating the amount of traffic on a node. In TAOF [10], Packet Transmission Rate

(PTR), which is similar to PDR, is used to represent the number of packets each node forwarded during a certain time period. The third metric is residual energy [11,12], which is used to indicate the network lifetime. For example, the work in [12] presents two different ways of estimating the residual energy of nodes to calculate a lifetime cost metric. Expected Lifetime (ELT) metric is proposed in [13] to estimate the time before a node dies if it keeps on forwarding the same quantity of traffic. The last metric is the number of children, which is used to evenly distribute child nodes to parent nodes to avoid overloaded parents. For example, in work [14], the number of child nodes associated with parent nodes is used. However, the number of children may not be able to indicate the traffic load at each node because each node may generate or forward different amount of traffic flow.

In addition, some implicit congestion detection methods have been proposed. For example, multipath routing is used as a new routing metric to find the bottlenecks in [13,15]. Recent works such as CoAR [16] and RPL-CGA [17], combine multiple routing metrics to address the load-balancing problem. However, how to combine multiple routing metrics without incurring much complexity is still an open problem.

Congestion Notification. There are several ways to notify congestion. One way is to send routing metrics in the optional Metric Container within the DIO message. The majority of previous works use this method. For example, in [3], a queue utilization is sent via a DIO message. In M-RPL [15], a DIO message is used to send packet delivery ratio (PDR) to parent nodes. The second method is to add a congestion notification bit in a DIO message header, as shown in CoAR [16]. The work in [18] proposed another method to indicate the congestion by delaying the transmission of DIO messages.

3 Frequent Parent Changes

When using a congestion-aware routing metrics introduced in the previous section to achieve load balancing, frequent parent changes or parent oscillations may occur. Frequent parent changes can induce instability in the network and energy consuming [19]. In this section, we first present two typical scenarios to explain the parent change problem. Then, we study the impact of the issue on the stability and performance of RPL.

3.1 Scenarios

To better understand the parent change problem, we use several examples to demonstrate the problem. In those examples, we use a general overloaded indicator, which represents the number of packets sent by a node during a certain time period. In reality, the indicator could be replaced by a queue utilization or packet delivery ratio [13,8,4,9].

Scenario 1: The first type of parent changes occurs when a group of child nodes change their preferred parents simultaneously. In Fig. 1, suppose that initially the three nodes select node 2 as their parent, and each node's overloaded metric is 5. Suppose that the indicator value of node 2 is 20, which is the sum of all its children and its self-value. When a new candidate parent 3 becomes available with a value of 5, all the child nodes will switch to it. After that, suppose that node 2 broadcasts DIO messages with a new lower value (5) because all the previous child nodes already left it. The DIO messages will trigger the child nodes simultaneously switch from their preferred parent node 3 to node 2 again. Similarly, when those child nodes receive another DIO from node 3, they will change their parent back to node 3 again. This occurs repeatedly, resulting in parent oscillations.

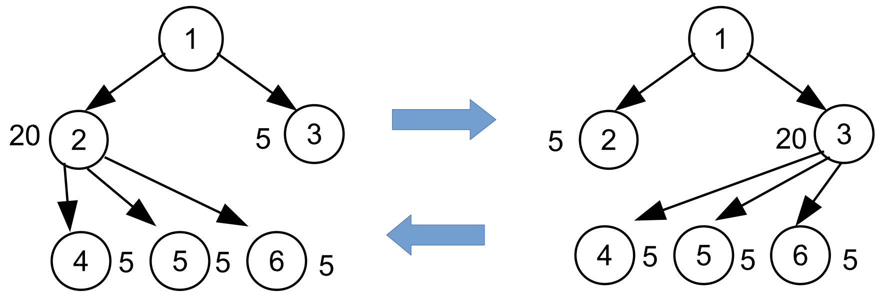


Fig. 1. An example of parent changes occurring at a set of nodes. The number beside a node represents the number of packets sent by the node during a period of time. A solid line indicates a preferred parent.

Note that this example is similar to the thundering herd problem [20,3]. The thundering herd problem is the case that when a new node joins a network with a small rank value, it may suddenly attract a large number of child nodes to switch their current preferred parents. However, the thundering herd problem may not necessarily cause parent oscillations because those children may not change their preferred parents after switching to the new parent.

This type of parent changes can be migrated by using a hysteresis threshold, a tolerance of small metric changes proposed in RFC 6719 [7]. Suppose that a node already selects a preferred parent. Then, another less loaded candidate parent appears. Before changing its preferred parent, the node compares the difference value of the metrics between the candidate and current parents. If the difference value is greater than a predefined hysteresis threshold, the node chooses the candidate parent as its new preferred parent. Otherwise, the node retains its current preferred parent. In addition, each child node may change its parents at different time [7], or stick with their current parents. In previous example, suppose that the threshold is set to 10. As shown in Fig. 2, after one of the child nodes switches from parent 2 to parent 3, the other nodes will stick with node 2 because the difference between the metrics of both parents (5) is less than the threshold. This example shows that a properly configured hysteresis threshold can reduce the occurrence of parent changes.

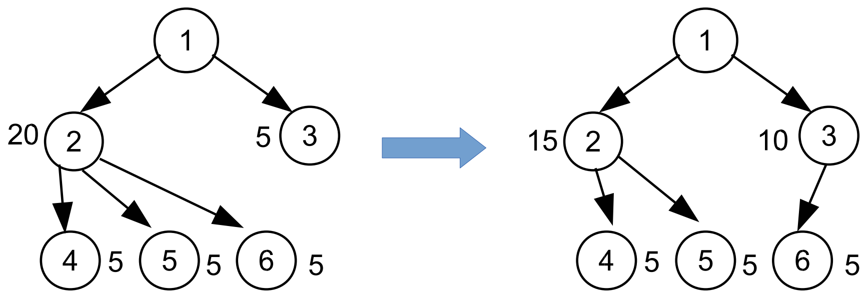


Fig. 2. An example of using a hysteresis threshold to migrate frequent parent changes. The hysteresis threshold is set to 10.

Scenario 2: Differing with the Scenario 1, the second type of parent changes occurs when some but not all child nodes change their preferred parents. The worst case is when some nodes with large traffic repeatedly switch their preferred parents. For example, in Fig. 3, there are four child nodes, and one of them, node 7, sends large traffic. Note that the node could be an IoT device generating large traffic, such as a digital camera, or a hot spot with many children. In this example, suppose that parent 2 is the preferred parent of other three nodes with light traffic, and node 7 selects parent 3 as its preferred parent. Let us assume that the hysteresis threshold is still 20. Thus, node 7 shall change its preferred parent to node 2 upon receiving a new DIO from node 2 because node 2 is less overloaded than parent 3. However, after switching to the new parent 2, node 2 becomes congested. Then, node 7 changes its preferred parent from 2 to 3 again, resulting in a parent oscillation.

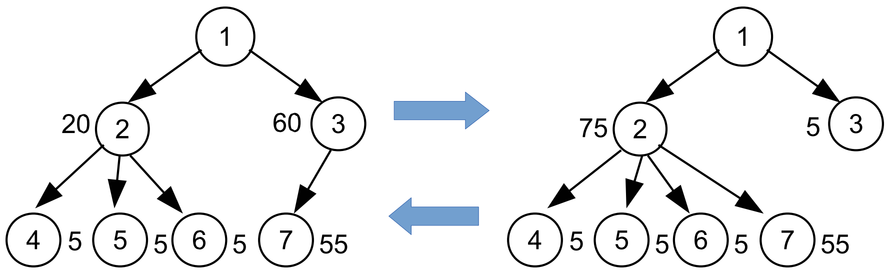
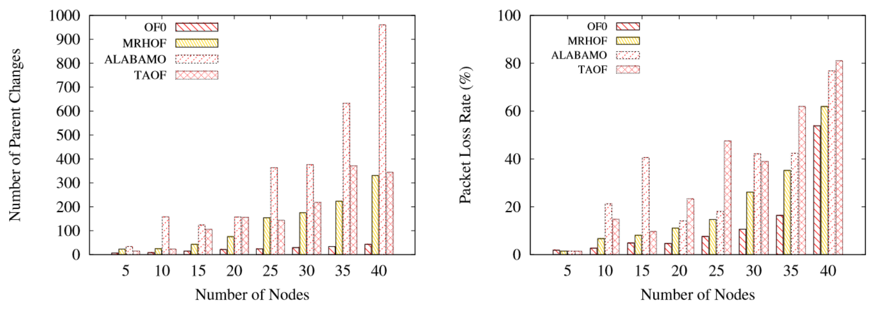


Fig. 3. An example of parent oscillations occurring at node 7.

A straight solution is to increase the hysteresis threshold for node 7. For example, if we increase the hysteresis threshold from 20 to 50, the oscillation will be avoided. However, using a large hysteresis threshold could cause an unbalanced IoT network. Another solution is to use different hysteresis thresholds at different nodes. However, it is hard to determine which threshold value is appropriate for each device, especially in a heterogeneous IoT network.



(a) Parent Changes (b) Packet Loss Rate

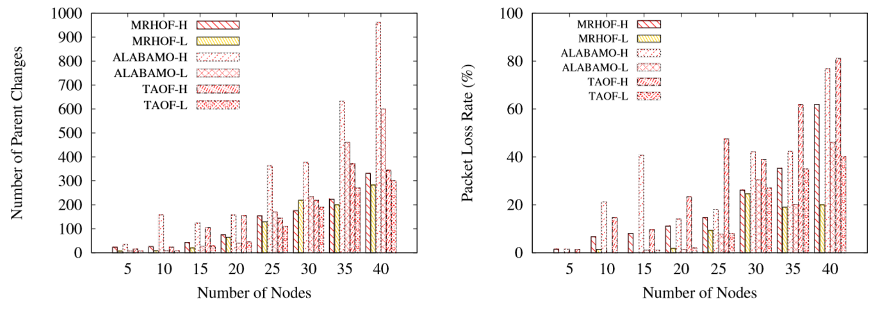
Fig. 4. The number of parent changes and packet loss rate for different network sizes.

3.2 Impact on Stability

In this subsection, we use simulation experiments to investigate frequent parent changes for different OFs and the impact of the problem on the stability of RPL. Each measurement is simulated in Cooja. We construct a cube topology, which consisted of nodes on a square grid where each node had no more than 8 neighbors. Each network topology consists of different number of sender nodes and one root node. We randomly select two nodes sending 30 packets per minute (30 ppm), and the rest of nodes with 5 ppm. Each test is simulated for 20 minutes. The details of the simulation setup are presented in Section 5.

We first investigate the number of parent changes and UDP packet loss rate for four different OFs: OF0, MRHOF, ALABAMO [9] and TAOF [10], under the same settings. Because OF0 and MRHOF do not consider load balancing, we use them as the baseline. The result in Fig. 4 exhibits high instability. That is, most of the nodes change frequently their parents. For example, for ALABAMO, more than half of the nodes change at least 20 times their preferred parent. Meanwhile, we do not find parent oscillations when OF0 is used. As a result, the packet loss rate is always lower than the other cases. However, when the network consists of 40 nodes, the packet loss rate is over 50%. That means, with the growth of network size, more traffic flows are inject into the network and lead to congestion. Since the whole network is unbalancing, some nodes could be overloaded. From Fig. 4(b), the packet loss rate is also very high when either ALABAMO or TAOF is used.

Next, to understand the impact of the nodes with large traffic on parent changes, we first set all nodes in the simulations with the same packet generation rate (0.5ppm). Then, we randomly choose two nodes to increase their rates to 30ppm. We compare the number of parent oscillations and UDP packet losses under the two configurations. Since OF0 does not have many parent oscillations, we do not consider it in the test. The results are shown in Fig. 5. We find that the scenarios with two nodes with higher packet generation rate typically cause more parent changes and more packet losses. We also observe that when the number of nodes is over 30, the number of parent changes in the configuration with the same generation rate is also very high. The reason is that some nodes close to the root aggregate packets from their child nodes so that they could forward a large number of packets. As a result, this configuration could also cause more parent changes.



(a) Parent Changes (b) Packet Loss Rate

Fig. 5. Comparison of number of parent changes and packet loss rate for two different configurations. An OF with “H” represents a simulation topology where two nodes generates more packet than others, while “L” represents a simulation topology where every node generates the same number of packets.

4 SL-RPL Design

SL-RPL employs a novel objective function for RPL that aims to migrate the instability due to frequent parent changes. The basic idea of SL-RPLis to provide a method so that every node can estimate the impact of its traffic on the parent’s overloaded indicator.

4.1 Parent Selection

To directly indicate the load of each node, SL-RPL uses two metrics: PTR and ETX, where PTR represents the number of packets each node transmitted during a certain time period. We calculate PTR by using a similar method proposed in [10]. More specifically, each node use a buffer to record the timestamps for each data packet transmitted. A node calculates a PTR value by comparing the timestamps of packets with the current time. ETX between node k and its parent node p_k is measured as:

$$ETX(k, p_k) = \frac{\text{Total transmissions from } k \text{ to } p_k}{\text{Successful transmissions from } k \text{ to } p_k} \quad (1)$$

In SL-RPL, PTR is added to the DIO message in the Metric Container. When a node joins a DODAG, its PTR may not be available. In this case, each node uses a default PTR value as its estimation. Differing with PTR, ETX does not need to broadcast to neighbors, and is used to estimate the expected PTR.

In addition, SL-RPL uses hop counts as a filter to identify a potential parent set. Specifically, the hop count of node k is defined as:

$$h(k) = h(p_k) + 1 \tag{2}$$

where $h(k)$ is the hop count between node k and its preferred parent p_k. The hop count metric is employed to filter out parent candidates that are far away from the root based on a predefined threshold.

To migrate the frequent parent changes, a child node calculates the PTR value of the current preferred parent by removing the expected PTR sent by itself. The expected PTR sent by a node k to its preferred parent p_k is defined as:

$$exp_k(PTR(p_k)) = \frac{PTR(k)}{ETX(k, p_k)} \tag{3}$$

Hence, the estimated PTR of a preferred parent after removing the expected PTR sent by node k is:

$$ePTR(p_k) = PTR(p_k) - exp_k(PTR(p_k)) \tag{4}$$

From a parent candidate set, a node chooses the preferred parent with the least estimated PTR value. In addition, the new parent candidate is compared with the previous parent to decide if the current parent should be changed. More specifically, the difference between their estimated PTR values is compared to a pre-defined hysteresis threshold. As a result, SL-RPL can allow the nodes with large traffic to stick with their parents during DODAG construction. At the same time, the nodes with light traffic can still achieve a fine-grained load balancing based on the selected metric and the hysteresis threshold.

4.2 Example

Finally, we use an example shown in Fig. 6 to demonstrate SL-RPL. Suppose that the hysteresis threshold is set to 10 for all nodes. To simplify this example, we assume that the ETX value of each node is 1, which means all the packets sent by a child node can be successfully received by its parent. In reality, the value is always lower than 1. In Fig. 6(a), at node 7, the expected PTR value for parent 2 is $75 - 55 = 20$. Thus, for node 7, the difference between parent 2 and 3 is $20 - 5 = 15$ so that the node cannot switch to parent 3. However, the other nodes can change to parent 3. For example, for node 4, the expected PTR is $75-5 = 70$. As shown in Fig. 6(b), node 4 can switch to parent 3. Finally, all the other nodes except node 7 can switch to parent 3, as shown in Fig. 6(c).

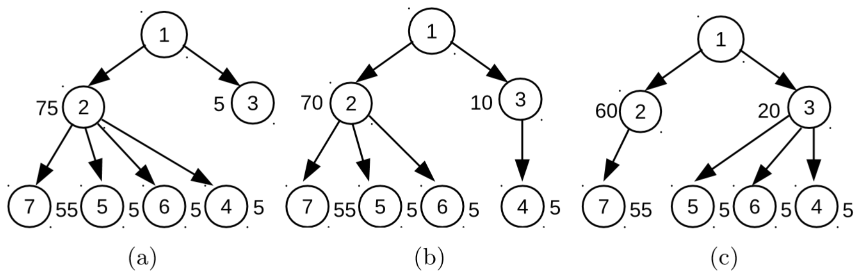


Fig. 6. An example of SL-RPL reducing the occurrence of parent oscillations at node 7.

5 Performance Evaluation

In this section, we evaluate the performance of SL-RPL. Our proposed method is tested using the popular simulator Cooja with Contiki OS. Cooja is a open source discrete network simulator to study the performance of RPL based IoT networks. The performance evaluation of SL-RPL was completed using the Cooja simulator provided by Contiki OS. The Z1 node type is used with a transmission range of 50 meters. We construct a cube topology, which consisted of nodes on a square grid where each node has no more than 8 neighbors within the range with cases between 5 nodes and 40 nodes in increments of 5. We setup our simulations by using the Unit Disk Graph Medium (UDGM) Distance Loss radio medium and a mote startup delay set as 1 second. Then, we continue the simulation to allow UDP messages to be propagated from child nodes to the root. The UDP packet interval of each node varies from 2 to 12 seconds (i.e., 30 to 5 ppm).

The performance of SL-RPL is compared against the performance of Hop Count (OF0), ETX (MRHOF), ALABAMO[1] and TAOF (which uses PTR). Improved network performance is evaluated for the number of preferred parent changes, the frequency of parent oscillations, packet loss rate, and power consumption.

The first measured performance metric is the number of preferred parent changes. As shown in Fig. 7, SL-RPL improves significantly the network stability. SL-RPL generates a little more parent changes than OF0. More than 80% of the nodes change less than 2 times. The second measured performance metric is the number of parent changes at each node. As shown in Fig. 8, using SL-RPL, most nodes only change twice, and some modes may change three times. On the contrary, the other OFs cause frequent parent changes at most of nodes. Next, we measure the packet loss ratio in Fig. 9. We find that SL-RPL can reduce the packet loss significantly. Finally, we measure power consumption during the whole simulation period. The power consumption is calculated by tracking the fraction of time that a node's CPU power mode. The total power consumption is to multiply the total CPU time with the voltage of the system and current consumption. Fig. 10 represents the average energy during the whole simulation against

[1] This code was obtained from research conducted by the University of Southern California's Autonomous Networks Research Group, http://anrg.usc.edu

the network size. We find that SL-RPL improves the energy efficiency and consumes less energy than other OFs. Our experiments show that SL-RPL clearly outperforms the standard RPL, ALABAMO and TAOF. It helps to migrate frequent parent changes with the consideration of load balancing.

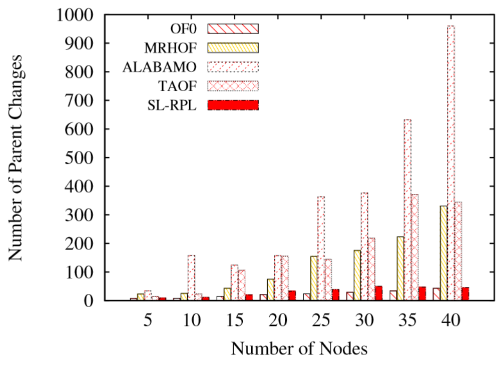


Fig. 7. Number of parent changes of OFs vs. the network topology size.

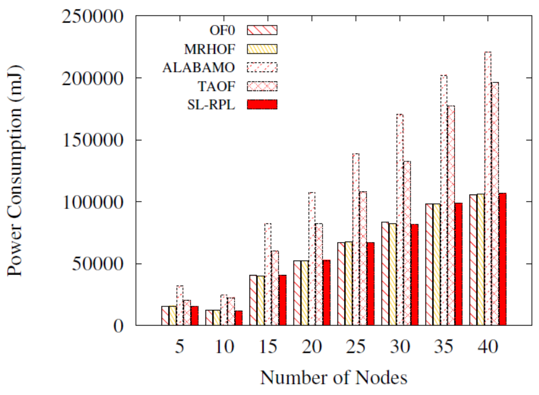


Fig. 8. Number of parent changes of each node vs. the network topology size.

6 Conclusion

In this paper, we propose a stability-aware load balancing mechanism to mitigate the frequent parent change issue of RPL. We compare the performance of SL-RPL with the standard RPL, ALABAMO and TAOF. Based on our simulation measurements, SL-RPL can clearly migrate frequent parent changes and provide load balancing for RPL.

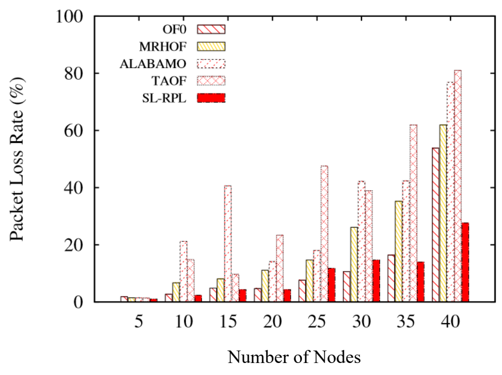


Fig. 9. UDP packet loss rate vs. the network topology size.

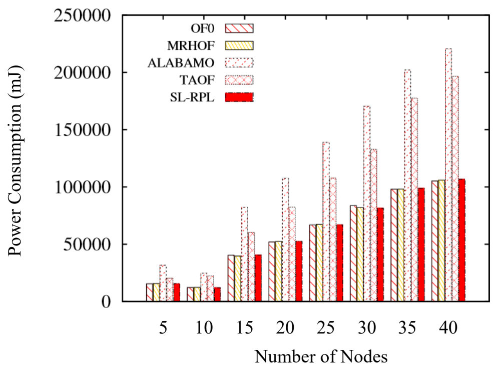


Fig. 10. Power Consumption vs. the network topology size.

References

1. Karkazis, P., Leligou, H. C., Sarakis, L., Zahariadis, T., Trakadas, P., Velivassaki, T. H., Capsalis, C. : Design of Primary and Composite Routing Metrics for RPL-compliant Wireless Sensor Networks. In: 2012 International Conference on Telecommunications and Multimedia (TEMU), pp. 13-18, Chania (2012).
2. Gaddour, O., Koubaa, A., Baccour, N., Abid, M. : OF-FL: QoS-aware fuzzy logicobjective function for the RPL routing protocol. In: 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 365-372, (2014).
3. Kim, H.-S., Kim, H., Paek, J., Bahk, S. : Load Balancing Under Heavy Traffic inRPL Routing Protocol for Low Power and Lossy Networks. In: IEEE Transactions on Mobile Computing. vol. 16, pp. 964-979, (2017).
4. Yang, X., Guo, J., Orlik, P. V., Parsons, K., Ishibashi, K.: Stability Metric BasedRouting Protocol for Low-power and Lossy Networks. In: 2014 IEEE International Conference on Communications (ICC), pp. 3688-3693, (2014).
5. Alexander, R. : RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,In: RFC 6550, (2012).

6. Thubert, P. : Objective Function Zero for the Routing Protocol for Low-Power andLossy Networks (RPL). In: RFC 6552, (2012).
7. Gnawali, O., Levis, P. : The Minimum Rank with Hysteresis Objective Function.In: RFC 6719, (2012).
8. Kim, H. : QU-RPL: Queue Utilization Based RPL for Load Balancing in LargeScale Industrial Applications. In: 12th Annual IEEE International Conference on Sensing, Communication and Networking, pp. 265-273, (2015).
9. Oliveira, T. B. : ALABAMO: A LoAd BAlancing MOdel for RPL. In: BrazilianSymposium on Computer Networks and Distributed Systems (SBRC), (2016).
10. Ji, C. : TAOF: Traffic Aware Objective Function for RPL-based Networks. In: 2018 Global Information Infrastructure and Networking Symposium (GIIS), pp. 1-5, (2018).
11. Iova, O. : Improving the network lifetime with energy-balancing routing: Application to RPL. In: 7th IFIP Wireless and Mobile Networking Conference, pp. 1–8, (2014),
12. Capone, S. : An Energy Efficient and Reliable Composite Metric for RPL OrganizedNetworks. In: Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing, pp. 178-184, (2014).
13. Iova, O. : Using Multiparent Routing in RPL to Increase the Stability and theLifetime of the Network. In: Ad Hoc Netw. vol. 29, pp. 45-62, (2015).
14. Qasem, M. : A New Efficient Objective Function for Routing in Internet of ThingsParadigm. In: 2016 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1-6, (2016).
15. Lodhi, M. A. : Multiple Path RPL for Low Power Lossy Networks. In: 2015 IEEEAsia Pacific Conference on Wireless and Mobile (APWiMob), pp. 279-284, (2015).
16. Bhandari, K. S. : CoAR: Congestion-Aware Routing Protocol for Low Power andLossy Networks for IoT Applications. In: Sensors, vol.18-11, pp.3838, (2018).
17. Cao, Y. : A Novel RPL Algorithm Based on Chaotic Genetic Algorithm. In: Sensors, vol.18-11, pp. 3647, (2018).
18. Liu, X. : Load Balanced Routing for Low Power and Lossy Networks. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 22382243, (2013).
19. Iova, O. : Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks. In: 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 2098-2102, (2013).
20. Hou, J. : Optimization of Parent-Node Selection in RPL-Based Networks. In: drafthou-roll-rpl-parent-selection-00, Internet Draft, (2017).