

# Sharing is caring: a cooperation scheme for RPL network resilience and efficiency

Brandon Foubert, Julien Montavont

► **To cite this version:**

Brandon Foubert, Julien Montavont. Sharing is caring: a cooperation scheme for RPL network resilience and efficiency. ISCC 2019 - 24th Symposium on Computers and Communications, Jun 2019, Barcelona, Spain. hal-02095410

**HAL Id: hal-02095410**

**<https://hal.inria.fr/hal-02095410>**

Submitted on 2 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Sharing is caring: a cooperation scheme for RPL network resilience and efficiency

Brandon Foubert  
Inria Lille - Nord Europe  
Villeneuve d'Ascq, France  
brandon.foubert@inria.fr

Julien Montavont  
ICube Laboratory (UMR CNRS 7357)  
University of Strasbourg, France  
montavont@unistra.fr

**Abstract**—The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) builds a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at one node. This node may act as a border router to provide Internet connectivity to the members of the DODAG but such a situation creates a single point of failure. Upon border router failure, all nodes connected to the DODAG are affected as all ongoing communications are instantly broken and no new communications can be initiated. Moreover, nodes close to the border router should forward traffic from farther nodes in addition to their own, which may cause congestion and energy depletion inequality. In this article we specify a full solution to enable border router redundancy in RPL networks. To achieve this, we propose a mechanism leveraging cooperation between colocated RPL networks. It enables failover to maintain Internet connectivity and load balancing to improve the overall energy consumption and bandwidth. Our contribution has been implemented in Contiki OS and was evaluated through experiments performed on the FIT IoT-LAB testbed.

**Index Terms**—Internet of Things, 6LoWPAN, RPL, cooperation, multihoming, failover, load balancing

## I. INTRODUCTION

The recent years have seen the introduction of everyday objects equipped with computation and communication capabilities. However, those objects (hereafter called nodes) have limited computational power, memory and energy due to their reduced size and cost. Nevertheless, such nodes can be networked together, forming a Low-power and Lossy Network (LLN). LLNs can benefit from a global Internet connectivity, building what is known as the Internet of Things (IoT). The IoT has numerous applications, such as environmental monitoring, home automation, medical uses, and many more.

The TCP/IP network stack was not designed to take energy or computational power constraints into account, which is why the IEEE and the IETF specified new standards adapted to the characteristics of IoT networks. Particularly, the proposed routing protocol is the IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [1]. RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) rooted at one node. This root node usually acts as a Border Router (BR) and relays communication between the LLN and the Internet.

In RPL networks, the connectivity of the entire LLN is directly dependent on the status of the BR. If the BR fails (e.g. hardware failure or heavy congestion), the whole LLN is disconnected from the Internet. The use of multiple BRs can mitigate such issue, but redundancy is expensive. Based

on recent studies such as [2], our feeling is that IoT deployments will most likely result in dense network colocalization, especially considering urban environments. Hence, this work presents a plausible way to take advantage of the BR redundancy offered by IoT networks colocalization.

In this article, we propose a solution to BR failure by enabling cooperation between colocalized RPL networks. Our contribution allows multiple RPL networks to dynamically discover each other and build a common DODAG. Upon BR failure, the remaining BRs can then automatically take over the traffic. In addition, a load balancing mechanism is running on BRs to alleviate the congestion caused by the DODAG structure, known as the funneling effect. At the network layer, seamless transition of nodes from one BR to another is enabled using the IPv6-to-IPv6 Network Prefix Translation (NPTv6) protocol [3]. Our proposal, referred to as RPL-NPT-LB, was implemented in Contiki OS and evaluated through experimentation on a real testbed. Results show that RPL-NPT-LB distributes evenly the traffic load between BRs and reduces transmission errors between nodes by up to 60%.

The rest of this paper is organized as follows: Section II introduces RPL and the issues related to its routing structure. Section III presents related works, followed by the description of our contribution RPL-NPT-LB in Section IV. Experimentation settings and results are detailed in Section V. Finally, conclusions and future works are presented in Section VI.

## II. PROBLEM STATEMENT

The DODAG root starts building the DODAG by periodically multicasting DODAG Information Object (DIO) messages. Those messages contain all the necessary parameters for surrounding nodes to join the DODAG. Upon reception, each node selects among candidate neighbors one or more preferred parents, *i.e.* preferred next hops toward the DODAG root, and may thereafter advertise the DODAG by sending its own DIOs. Nodes can also request immediate DIO transmission by sending a DODAG Solicitation Object (DIS) message. An example of a DODAG is depicted in Fig. 1.

Once a node has joined the DODAG, it can send a Destination Advertisement Object (DAO) message to build a downward route to itself. Downward route construction has two modes of operation: storing and non-storing. In storing mode, all nodes maintain a routing table and send messages

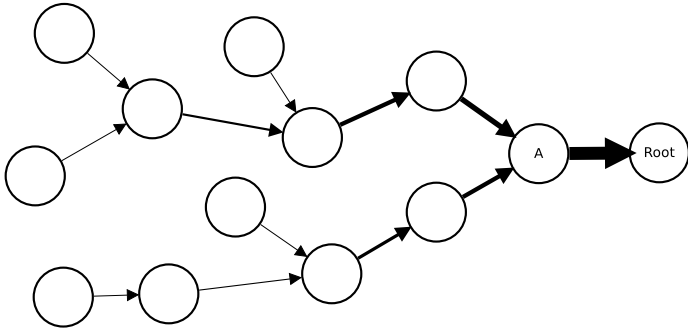


Fig. 1. DODAG ex. - the bigger the arrow, the more traffic the link handles

according to it. The messages are then forwarded in a hop-by-hop fashion. Using non-storing mode, only the DODAG root maintains a view of the whole network. The messages are first forwarded to the root and then source-routed to their final destination.

RPL routing structure may suffer from two main issues: border router failure and the funneling effect. On one hand, BR failure could be an electronic or electrical failure, a software bug as well as a heavy congestion leading to an irreversible overload, preventing all nodes in the LLN to communicate with the Internet. Moreover, in non-storing mode local communications (between nodes of the same LLN) are source-routed by the BR. Hence, data packets will be delayed or even dropped which puts time or loss sensitive applications (e.g. healthcare monitoring) at risk. On the other hand, with RPL a node should forward all the traffic from its sub-DODAG toward the BR, on top of its own traffic. The closer a node is from the BR, the more likely it is to have many children in its sub-DODAG. This phenomenon is known as the funneling effect and can be observed in Fig. 1. Nodes suffering from it (e.g. node A) may prematurely run out of energy, disconnecting their sub-DODAG from other nodes.

### III. RELATED WORK

The negative effects of BR failure and the funneling effect could be solved by introducing BR redundancy in the LLN, as nodes could have multiple exit points toward the Internet. RPL supports the use of multiple BRs with the concept of virtual DODAG root. The idea is to have multiple DODAG roots acting as standard nodes and pretending to be connected to a unique DODAG root. However, the RPL standard does not specify how to implement such a feature. Nevertheless, providing a RPL network with multiple BRs has been studied in the scientific literature.

In [4, 5, 6], several RPL-based routing protocols using multiple roots are designed and evaluated through simulations. Obtained results show that increasing the number of roots improves the overall performance of the network. Those works however do not consider how remaining roots can take over if one of them fails.

An implementation of multiple LoWPAN border routers is presented in [7]. Here, a single root BR maintains the network,

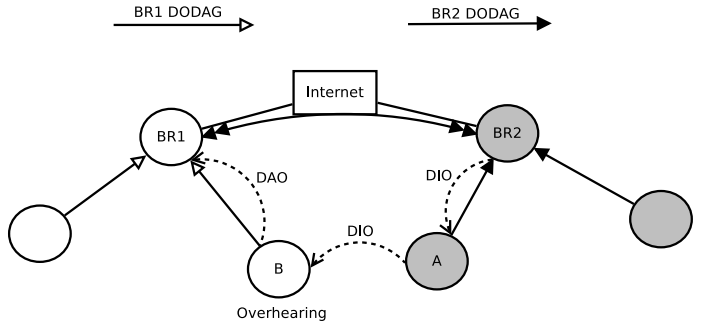


Fig. 2. Discovery ex. - node A overhears node B and forwards the information

while other BRs act as standard nodes and forward the traffic to the main root via an outer network (e.g. the Internet). In [8], multiple BRs are connected to a common backbone network, into which is located a unique RPL root (i.e. the root is outside the LLN). In [9], the authors propose a mechanism to support multiple roots in RPL based on a virtual DODAG root. A unique central unit known as the registrar synchronizes DODAG parameters for all root nodes, thus creating a virtual DODAG root, which is then maintained by the registrar. The work presented in [10] proposes a similar approach to create a virtual DODAG root in a RPL network. All those works suffer from the same weakness: the use of a unique central device to maintain a virtual DODAG root. Using a central device on which depends the proper functioning of routing operations just shifts the single point of failure from the DODAG root to this device.

The authors of [11, 12] try to limit the funneling effect by allowing nodes to choose their preferred parents based on the load of the candidate neighbors. But as the node redirection is a local decision and is not coordinated, such solution may prevent RPL from converging, resulting in an unstable network. In [13], a mechanism leveraging multiple cooperative RPL instances based on game theory is proposed. However this work does not consider BR failure nor does it provide a failover mechanism.

### IV. RPL-NPT-LB

Our contribution, called RPL with Network Prefix Translation and Load Balancing (RPL-NPT-LB), is based on the assumption that multiple RPL networks, belonging to different operators, are colocalized. RPL-NPT-LB is composed of three parts: a distributed virtual DODAG root, a dynamic load-balancing mechanism and an IPv6 network prefix translation.

#### A. Virtual DODAG root

The concept of virtual DODAG root was first introduced in [1], but this document did not fully specify how it should be implemented. A virtual DODAG root is obtained when multiple BRs from a unique RPL network coordinate themselves to act as if they were a unique DODAG root. This introduces BR redundancy without the need to modify deeply RPL operations. In contrast to [9, 10], our proposal allows colocalized BRs to dynamically discover each other as

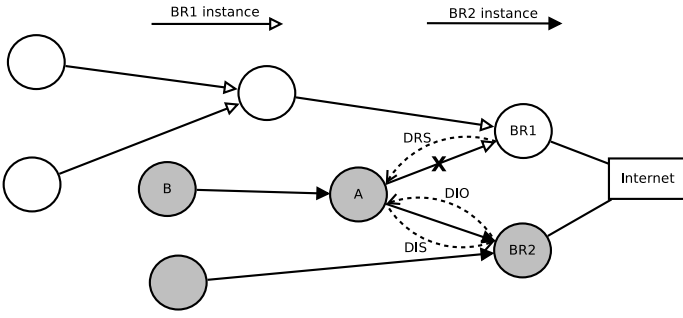


Fig. 3. Redirection ex. - node A is redirected by BR1 to BR2's instance

illustrated in Fig. 2. Once a node overhears a DIO from a foreign DODAG, it can piggyback the discovered information (e.g. the foreign BR IPv6 address) into a DAO sent to its own BR, similarly to what is proposed in [14]. Upon reception, the BR can contact the foreign BR and then synchronize on a common set of DODAG parameters.

After the discovery process, cooperating BRs build a common DODAG by maintaining a virtual DODAG root. It means that the cooperating BRs announce the same DODAGID and nodes joining this new DODAG only see one DODAG root.

### B. Load balancing

Load balancing is the action of evenly splitting the traffic load among multiple paths, aiming at avoiding bottlenecks and uneven workload distribution. In case of multiple BRs, one can make nodes (and their sub-DODAG) change their preferred parents to force the traffic originated from this part of the network to go through another BR. However, local decisions can cause jitter and suboptimal redirections. In RPL-NPT-LB, we specified a flexible mechanism to explicitly redirect nodes from a congested network to a less congested one. Redirections are carried out by the cooperating BRs in order to make decisions from a global network point-of-view.

The RPL specification defines a RPL instance as a set of one or more DODAG roots. We propose a different usage for the RPL instance: a common DODAG is going to contain one or more BRs coordinated to use different RPL instances. This allows us to differentiate which BR each node is attached to. BRs use the RPL non-storing mode, thus each node should send a DAO to its own BR. Nodes turn on a special flag in their DAOs when they see a node from a different instance in their vicinity (when receiving DIOs). This advertises them as *redirectable nodes* to their BR. After synchronizing the received information, all BRs know all existing destinations in the common DODAG and the set of *redirectable nodes*.

Cooperating BRs can enter in congested mode depending on a specific trigger (e.g. max sub-DODAG size, queue losses...). When a BR enters in congested mode, it selects a set of nodes to redirect from the set of redirectable nodes taking part in its instance. The selection is based on how many children and if it matches the number of nodes the BR wishes to redirect (e.g. if  $x$  nodes should be redirected, it tries to redirect one node with  $x - 1$  children). It then sends a new unicast RPL

TABLE I  
EXPERIMENTAL PARAMETERS

MAC layer	IEEE 802.15.4 CSMA/CA
MAC acknowledgments	Enabled
MAC Tx queue size	1 packet
RDC mechanism	No RDC (NULLRDC)
Traffic type	UDP packets
Traffic rate	20 bytes packet per second
Tx power	3 dBm
Rx power threshold	-60 dBm
Motes used	10 M3 open node
RPL mode	Non-storing
RPL OF	MRHOF
OF metric	ETX
Congested mode trigger	Sub-DODAG size threshold
Duration	60 minutes

message called a *DODAG Redirection Solicitation (DRS)* to the chosen nodes. Upon reception of a DRS from its BR, a node has to switch to a different instance from the one it is currently in. To this end, each node retains the IPv6 address of the best ranked node from a different instance in its reach.

Fig. 3 depicts an example of redirection in which node A is defined as a *redirectable node*. After a certain amount of time, BR1 enters in congested mode and chooses to redirect node A. It then sends a DRS to node A and deletes it from its sub-DODAG list. Upon DRS reception, node A enters in a transitioning redirection state. First it sends a unicast DIS to the retained IPv6 address from the other instance in order to receive a fresh DIO (stored information could potentially be obsolete at the moment of the DRS reception). Upon reception of a DIO from BR2, node A joins BR2's instance. Nodes in the sub-DODAG of A (node B in Fig. 3) automatically follow A in BR2's instance.

### C. IPv6-to-IPv6 network prefix translation

IoT networks are likely to be served by different Internet Service Providers (ISPs). This means that the IPv6 prefixes distributed in the LLNs by the BRs will be different too. A node should therefore change its IPv6 address when switching from one BR to another. This breaks ongoing communications and consumes time and resources for the new address configuration. To avoid IPv6 address reconfiguration, we adapted the IPv6-to-IPv6 Network Prefix Translation (NPTv6) [3] protocol. It allow BRs from different ISPs to distribute a default unique local IPv6 prefix [15] in the common DODAG. For outgoing packets, BRs will translate the local IPv6 prefix of the source address with the global one. Inversely, BRs will translate the global IPv6 prefix of the destination address with the local one for incoming packets. Thanks to provider-independent address space, cooperating BRs can share their global IPv6 prefixes and position themselves as backup routes for prefixes owned by colocalized BRs. Upon a BR failure, orphan nodes can join a neighbor BR while keeping the same IPv6 address (both inside and outside the LLN), therefore preserving their connections with hosts across the Internet.

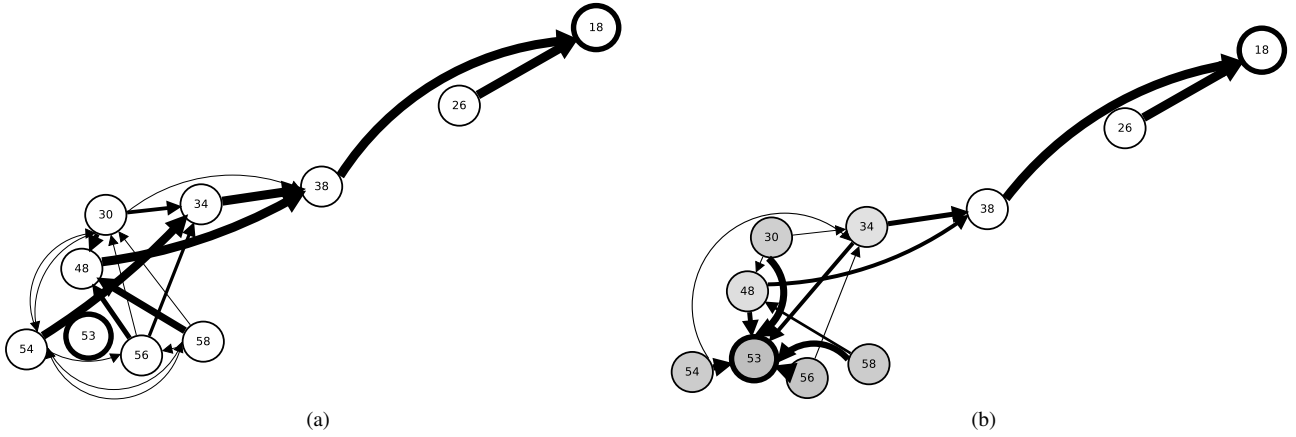


Fig. 4. RPL topologies and physical topology - 4a from RPL and 4b from RPL-NPT-LB, the bigger the arrow, the more frequently the link is used

## V. EXPERIMENTATION AND RESULTS

Our implementation is based on Contiki OS version 3.x. Contiki OS [16] is a lightweight open-source operating system for IoT devices. It comes with a full IPv6 network stack and an implementation of the RPL standard, known as Contiki RPL.

### A. Experiment settings

Our contribution, RPL-NPT-LB is implemented on top of Contiki RPL. We have conducted our experiments on a real testbed known as FIT IoT-LAB [17], to compare our proposal against RPL. It is a large scale testbed allowing computer scientists to experiment with constrained wireless networks. We used the M3 nodes provided in IoT-LAB, whose hardware capabilities can be found in [18]. Experimental parameters used for the experiments are regrouped in Table I.

We tested our proposal under a controlled topology to analyze more precisely the observed results. We used 10 M3 nodes: 2 BRs (nodes 18 and 53) and 8 nodes sending a regular amount of traffic destined to a remote host. Our scenario simulated the situation in which a new RPL network boots in the vicinity of a stable one. Therefore, BR 53 starts 60 seconds later than BR 18. The 8 other nodes send UDP packets at a rate of one per second. They start to transmit 30 seconds after the beginning of an experiment.

### B. Results

The results presented in this section are an average of 100 experiments, 50 using RPL and 50 using RPL-NPT-LB.

Fig. 4 shows the DODAG topologies formed by the nodes on top of the physical topology. Arrows represent the DODAG links: the thicker they are, the more frequently they are used. Fig. 4a is obtained with RPL while Fig. 4b is obtained with RPL-NPT-LB. The shading of each node represents at which frequency it is taking part in each instance: white is BR1's instance, gray BR2's. With RPL, we can see that BR 53 does not have any child node because it belongs to a different ISP. Because of this, node 38 has to relay the traffic from 6 children (nodes 30, 34, 48, 54, 56 and 58), which creates congestion.

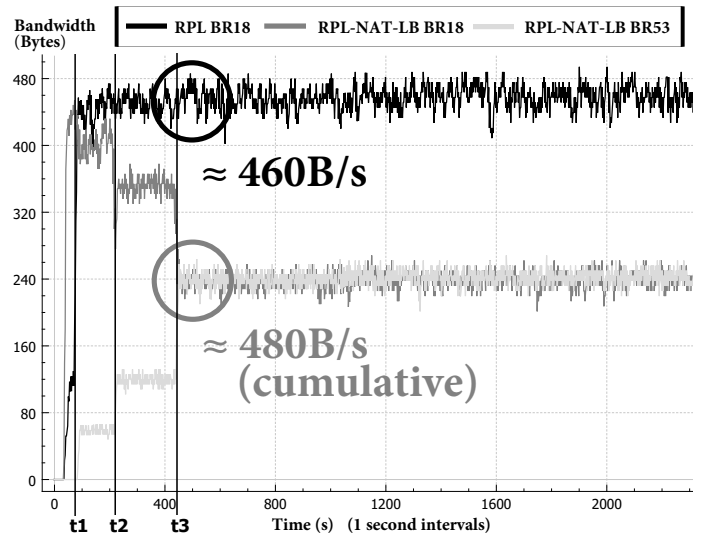


Fig. 5. Bandwidth repartition ex. - load balancing between the BRs

With RPL-NPT-LB, we observe an even distribution of the children among the two BRs.

Fig. 5 illustrates how the bandwidth is shared between the BRs in bytes per second as a function of time in two experiments, one with RPL and one with RPL-NPT-LB. Using RPL, BR 53 does not forward any traffic because it has no children. Considering RPL-NPT-LB, within the first 500 seconds of the experiment the scenario unravels as follows: BR 18 wakes up together with the 8 other nodes. All nodes join the DODAG and start to transmit UDP data toward a remote destination. BR 18 enters congested mode but no nodes are *redirectable* until the wake up of BR 53. Once BR 53 multicasts DIOs, neighbor nodes become *redirectable nodes*. BR 18 then sends a DRS at  $t_1$ , and a second one at  $t_2$ . Then, at  $t_3$  BR 18 redirects two nodes with one DRS. From this point forward, the load is equally balanced between both BRs. The mean cumulative bandwidth for both BRs is 460 bytes per second using RPL, and 480 bytes per second using RPL-NPT-

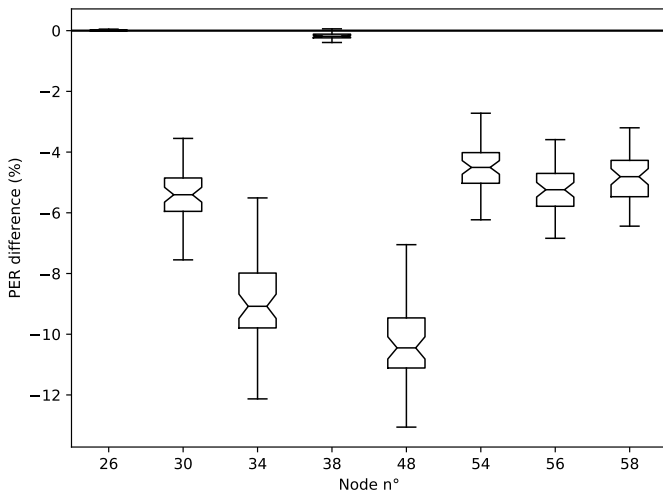


Fig. 6. End-to-end packet error rate difference per node

LB. This is due to a lower amount of packet losses when using RPL-NPT-LB.

The Packet Error Rate (PER) is the ratio between the number of lost packets and the total number of sent packets. The end-to-end PER of a node represents the ratio of its lost packets on the path from the node to the packet's destination. Fig. 6 shows the end-to-end PER difference per node between RPL and RPL-NPT-LB in percentage. It is the results of RPL minus the results of RPL-NPT-LB, which represents the proportion of loss we avoid with our proposal compared to RPL. As we can see, the end-to-end PER achieved by RPL-NPT-LB is generally better than the one presented by RPL. However, due to the 802.15.4 acknowledgments and retransmissions, the difference is only up to 12%. This observation can be explained by looking at the repartition of the transmission states.

In Contiki, each transmission can end with different states. The repartition of the MAC layer transmission states for each node is represented in Fig. 7. Each state that involves a transmission is counted, whether it is successful or not. Left columns are results obtained with RPL, and right columns are results obtained with RPL-NPT-LB. A transmission is either an acknowledgement (Ack tx) or something else (e.g. a data packet or a control message). If the transmission is not an acknowledgement, then it is a packet sent after which an acknowledgement should be received. If a valid acknowledgement is received within the timer window, then the transmission is valid (Tx ok). If an invalid acknowledgement (or anything else) is received instead, the transmission is invalid (Ack collision). If nothing is received before the timer ends, the transmission is also invalid (No ack). Finally, if a valid transmission is a data packet that must be forwarded, the forward queue of the receiving node may be full. In this case, the packet is dropped and the transmission lost (Queue drop). Overall, transmission problems experienced by nodes are largely reduced with RPL-NPT-LB thanks to our load

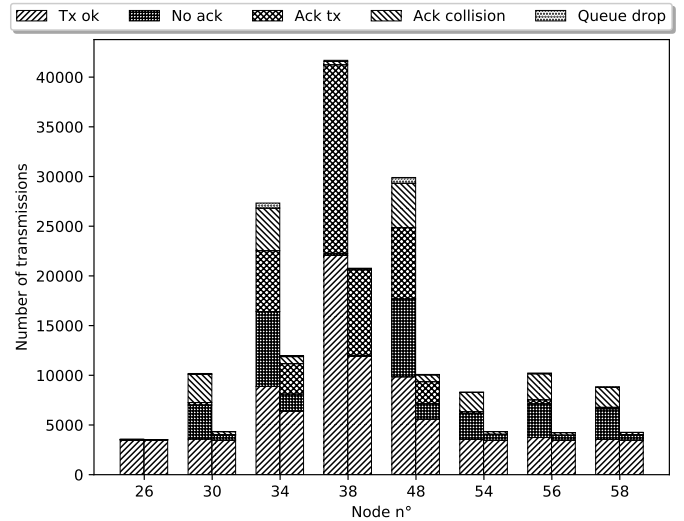


Fig. 7. MAC transmission status repartition per node

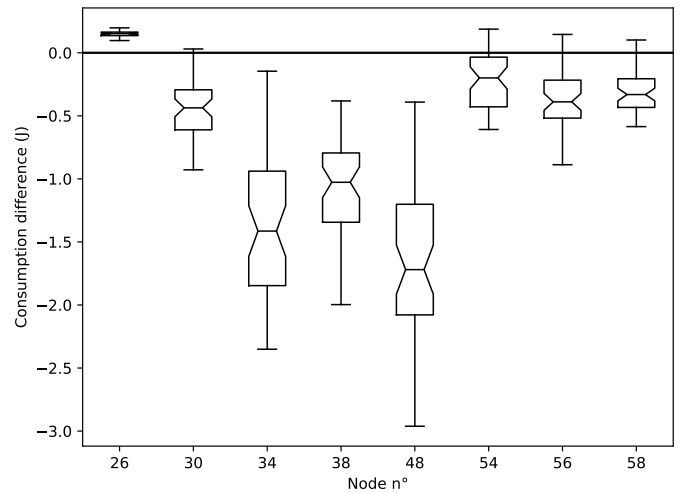


Fig. 8. Energy consumption difference per node

sharing mechanism. Nodes are efficiently redirected to less congested path, which reduces the amount of transmission errors. Although RPL achieves an end-to-end PER close to the one of RPL-NPT-LB, this came at the cost of many retransmissions at the MAC layer, especially for nodes 34, 38 and 48.

Increasing the number of retransmissions should also increase the energy consumption. Fig. 8 represents the energy consumption difference of the nodes between RPL and RPL-NPT-LB in joules at the end of the experiments. It represents the amount of energy consumed in one hour with our proposal compared to RPL. In RPL-NPT-LB, all nodes use less energy, except node 26. Such reduction is directly correlated with the reduction of the number of retransmissions at the MAC layer. Additionally, as the traffic is evenly split between the two BRs, the energy consumption is better balanced between

nodes. However, the energy consumption difference is not as significant as expected regarding Fig. 7. We chose to keep the nodes always on (NULLRDC in Contiki) to not introduce bias in the performance comparison. So, the energy spent for transmissions is shrunk by the one spent in reception due to the *always on* configuration. As illustrated in Fig. 7, node 26 does not experience transmission errors using RPL. Being a leaf node in the DODAG, it transmits the same amount of packets using RPL or RPL-NPT-LB. So, the slight difference in energy consumption came from the reception mode, which is not significant in our configuration.

## VI. CONCLUSION AND FUTURE WORK

In this article we proposed a full solution to support border router redundancy to mitigate the effects of BR failure and the funneling effect. BRs discover themselves and share information to establish a connection via the Internet. This allows the creation of a virtual DODAG that enables multiple BRs to be part of a common DODAG. The discovery and exchange of information are fully distributed. Then, an explicit redirection mechanism is used to redirect nodes from one BR to another, which evenly share the load in the network. Finally, we adapted the NPTv6 protocol to avoid IP address change upon redirection from different IPv6 networks.

Our proposal RPL-NPT-LB was implemented in Contiki OS. We assessed the efficiency of our protocol by conducting experiments on the FIT IoT-LAB testbed. Results show that the bandwidth is equally shared between the border routers. As a result, nodes experience less transmission errors which in turn reduces the end-to-end packet error rate together with the energy consumption.

Encouraged by the results presented here, we plan to extend our performance and scalability studies to large scale experiments. We intend to increase the number of nodes and BRs to reflect more complex scenarios, and compare RPL-NAT-LB with other works of the literature. We also want to study alternative triggers to switch a BR into congested mode such as packet losses or remaining energy. Finally, we are working on improving the selection process using specific metrics such as link quality, and we will investigate security issues coming from the cooperation between different networks.

## REFERENCES

- [1] T. Winter et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. Mar. 2012.
- [2] D. Evans. *The Internet of Things - How the Next Evolution of the Internet Is Changing Everything*. Tech. rep. Cisco - Internet of Things Group, April 2011.
- [3] M. Wasserman and F. Baker. *IPv6-to-IPv6 Network Prefix Translation*. RFC 6296. June 2011.
- [4] M. O. Farooq et al. "RPL-based routing protocols for multi-sink wireless sensor networks". In: *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2015, pp. 452–459.
- [5] M. O. Farooq and T. Kunz. "IoT-RF: A routing framework for the Internet of Things". In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Oct. 2017, pp. 1–7.
- [6] K. Nesrine et al. "Rescue-Sink: Dynamic sink augmentation for RPL in the Internet of Things". In: *Transactions on Emerging Telecommunications Technologies* 29.2 (2018).
- [7] W. Ge et al. "Implementation of multiple border routers for 6LoWPAN with ContikiOS". In: *2015 International Conference on Information and Communications Technologies (ICT 2015)*. Apr. 2015, pp. 1–6.
- [8] L. Deru et al. "Redundant Border Routers for Mission-Critical 6LoWPAN Networks". In: *Real-World Wireless Sensor Networks*. 2014, pp. 195–203. ISBN: 978-3-319-03071-5.
- [9] D. Carels et al. "Support of multiple sinks via a virtual root for the RPL routing protocol". In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (June 2014), p. 91. ISSN: 1687-1499.
- [10] Q. D. Nguyen et al. "RPL Border Router Redundancy in the Internet of Things". In: *Ad-hoc, Mobile, and Wireless Networks*. 2016, pp. 202–214. ISBN: 978-3-319-40509-4.
- [11] H. S. Kim et al. "Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks". In: *IEEE Transactions on Mobile Computing* 16.4 (Apr. 2017), pp. 964–979. ISSN: 1536-1233.
- [12] M. Ha et al. "Dynamic and Distributed Load Balancing Scheme in Multi-gateway Based 6LoWPAN". In: *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*. Sept. 2014, pp. 87–94.
- [13] M. Barcelo et al. "Cooperative interaction among multiple RPL instances in wireless sensor networks". In: *Computer Communications* 81 (2016), pp. 61–71. ISSN: 0140-3664.
- [14] M. M. Khan et al. "A multi-sink coordination framework for low power and lossy networks". In: *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*. Mar. 2016, pp. 1–5.
- [15] R. Hinden and B. Haberman. *Unique Local IPv6 Unicast Addresses*. RFC 4193. Oct. 2005.
- [16] A. Dunkels, B. Gronvall, and T. Voigt. "Contiki - a lightweight and flexible operating system for tiny networked sensors". In: *29th Annual IEEE International Conference on Local Computer Networks*. Nov. 2004, pp. 455–462.
- [17] C. Adjih et al. "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed". In: *IEEE World Forum on Internet of Things (IEEE WF-IoT)*. Dec. 2015.
- [18] *M3 open node hardware*. URL: <https://www.iot-lab.info/hardware/m3/>.