ODeSe: On-Demand Selection for multi-path RPL networks

Tomas Lagos Jenschke, Remous-Aris Koutsiamanis, Georgios Papadopoulos, Nicolas Montavont

ODeSe: On-Demand Selection for Multi-path RPL Networks

Tomas Lagos Jenschke[a,*], Remous-Aris Koutsiamanis[a], Georgios Z. Papadopoulos[a] and Nicolas Montavont[a]

[a]*IMT Atlantique, Irisa, France*

ARTICLE INFO

Keywords:
Industrial IoT
LLNs
RPL Routing Protocol
Reliable and Available Wireless (RAW)
PAREO Functions
Multi-path Routing
Determinism
IEEE Std 802.15.4-2015 TSCH

ABSTRACT

The Internet of Things (IoT) has become ubiquitous due to its flexibility, ease-of-use, and reduced cost. As a consequence, the industry is adopting these systems in its transformation into Industry 4.0. However, the strict Quality of Service (QoS) requirements of the industry are not met with the default best-effort provisions of the IoT. Most industrial applications require strict guarantees in terms of end-to-end network reliability and latency. For instance, consecutive packet losses can lead to communication disruptions in supply chain systems. Therefore, adaptations are being made to fulfill these requirements with the IEEE Std 802.15.4-2015 Time slotted Channel Hopping (TSCH) link-layer standard and the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) standard at the Internet Engineering Task Force (IETF). However, even by employing such industrial protocols, it is still difficult to achieve the expected QoS levels. Considering that RPL constructs and maintains a single-path from a source to a destination if there are potential issues on this path (e.g., queue overflow, variable wireless link quality) packets may suffer unexpected delays and even drops. If we consider a multi-path implementation where each node can replicate a packet into several paths, the transmission reliability improves since each packet copy is used to forward the packet information. However, uncontrolled replication can lead to network flooding, resulting in excessive power consumption. In this article, we present On-Demand Selection (ODeSe), a novel multi-path routing algorithm, which improves our previous work, the Common Ancestor (CA) algorithms, by selecting the most suitable upward forwarders. Moreover, we use the Packet Automatic Repeat reQuest, Replication and Elimination, and Overhearing (PAREO) functions in order to improve network reliability and availability. In order to control the number of relay nodes during a transmission, ODeSe forces each node of the same relay level to select the same Preferred Parent (PP) and the same Alternative Parent (AP). Thus, we address the trade-off between the total number of traversed nodes in the network and high reliability. Using the Cooja network simulator running the Contiki OS, we compare ODeSe against single-path RPL and multi-path RPL with different alternative parent selection algorithms. The results demonstrate that ODeSe outperforms single-path RPL in terms of reliability, and multi-path RPL in terms of energy consumption while maintaining a 99.14% packet delivery ratio.

1. Introduction

The importance of wireless technologies and the use of the pervasive Internet of Things (IoT) has dramatically risen in the last decade. The industry has started adopting these technologies in plants for different purposes, including monitoring and controlling production chains. The IoT is considered as one of the main pillars of what is called the Industry 4.0, the new industrial revolution guided by digitalization. The advantages of the IoT are its easy deployment, low energy consumption and great adaptability to different environments. One remaining issue though lies in wireless technology behavior and the industrial requirements, since IoT devices usually provide best-effort networking while Quality of Service (QoS) is required in industry applications.

The community has published several standards to handle these requirements. At the physical and link layers, the IEEE 802.15.4-2015 [1] standard is based on Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) designed for Low Power and Lossy Network (LLN) devices. As a part of IEEE 802.15.4-2015, the Time slotted Channel Hopping (TSCH) protocol allocates specific slots for transmission and reception per device while using frequency hopping for multi-path fading avoidance. At the network layer, a commonly used protocol for the IoT systems is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [2], standardized by the Internet Engineering Task Force (IETF). RPL configures one or more routes to transmit a message to a sink through the construction of a Destination Oriented Directed Acyclic Graph (DODAG). These protocols, however, do not provide a high level of quality of service since they depend on the variability of the link quality and the availability of the selected forwarders.

Multi-path routing protocols have been popular over the past years for different reasons, including to enhance reliability. They enable multiple paths using different nodes from a source to a destination to achieve different purposes, such as mitigating traffic load or improving transmission reliability. Depending on the use case, these multiple paths can be used alternatively or simultaneously.

In our previous work [3], we proposed the use of a braided pattern by simultaneously sending multiple copies of a packet to both a primary and a secondary route. This pattern allows receivers to overhear a transmission intended for another forwarder in order to increase the probability of receiving the transmitted packet. Here the problem lies in the selection of these Alternative Parents (APs) since this algorithm does not

*Corresponding author
tomas.lagos-jenschke@imt-atlantique.fr (T. Lagos Jenschke); remous-aris.koutsiamanis@imt-atlantique.fr (R. Koutsiamanis); georgios.papadopoulos@imt-atlantique.fr (Georgios Z. Papadopoulos); nicolas.montavont@imt-atlantique.fr (N. Montavont)
ORCID(s):

maintain a stable braided pattern [4], producing unnecessary routing changes within the DODAG.

In this work, we propose the On-Demand Selection (ODeSe) algorithm, which aims to perform multi-path parent selection based on the dynamic conditions at the packet-forwarding time. The aim is to keep a more stable braided pattern in the routing topology and prevent the forwarded packet from spreading to a larger part of the topology. This work also implements and uses the Packet Automatic Repeat reQuest, Replication and Elimination, and Overhearing (PAREO) [5] functions, to improve reliability and increase energy efficiency. Finally, in this work we use a centralized TSCH schedule for simplicity but the proposed algorithms are more general since RPL is agnostic of the Medium Access Control (MAC) layer.

The document is organized as follows: in Section 2 the TSCH and RPL protocols are explained. In Section 3 we introduce multi-path routing and the PAREO functions. Next, we provide a detailed description of the method used for implementing the braided routing pattern in Section 4. In Section 5 we present the proposed algorithms for parent selection. In Section 6 we describe the simulation setup used for our experimental evaluation and in Section 7 we analyze and present the experimental results. After having presented our contributions, we present other related works and how they compare to this work in Section 8. Finally, in Section 9 we conclude our work.

2. Background

2.1. IEEE Std 802.15.4-2015 TSCH

IEEE Std 802.15.4-2015 Time slotted Channel Hopping (TSCH) [6] is a wireless MAC protocol that provides predictable medium access and mitigates potential multi-path fading. TSCH allows nodes in the network to coordinate through a communication scheduler which uses TDMA for the time dimension and FDMA for the frequency dimension.

More specifically, time is divided into *timeslots* and the available frequency range into multiple non-overlapping physical radio *channels*. We call *cell* a (timeslot, channel offset) pair, which is the representation of a communication opportunity at a specific point in time and frequency. Frequency Hopping Spread Spectrum (FHSS) is employed to mitigate multi-path fading and for this reason the cells in the schedule do not use the physical radio channel values directly but a *channel offset*, which is mapped at each timeslot to a different but deterministically computed physical channel. Finally, a set of timeslots are organized into a *slotframe* of length equal to the repetition period (in timeslots). IEEE Std 802.15.4-2015 describes how TSCH should operate, but the definition of the schedule is left to specific implementations based on the needs of the application domain.

2.1.1. Cell types

As shown in Figure 1, it is possible to classify cells into two types.

- Shared cells: these are the contention-based cells, i.e., Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and they are used for control plane traffic.
- Dedicated cells: these are the contention-free cells. Only a specific pair of nodes can communicate concurrently so collisions are avoided within a given TSCH network. These cells are commonly used for application traffic.

2.1.2. Channel offset mapping

To support FHSS, a channel offset is mapped to a physical channel using the following Equation (1):

$$CH = F[(\text{ASN} + \text{ChannelOffset}) \mod n_{\text{CH}}] \quad (1)$$

where n_{CH} is the number of existing physical channels, Absolute Slot Number (ASN) is the timeslot counter for the lifetime of the network, and F is a lookup table that maps the results from the equation to the radio channel CH.

2.1.3. Scheduling

Regarding the scheduling, we use the scheme provided by TSCH to define the general structure and operation of the schedule. However, cell assignment, creation, and distribution of the schedule among the nodes are left to the implementation. These latter processes [7] can be classified into two broad categories:

- Centralized scheduling: a schedule is computed in a single central entity, taking into account the needs of all the nodes of the network, and is distributed to the nodes afterwards.
- Decentralized scheduling: nodes negotiate and create their schedule with the other nodes in their neighborhood. Thus, different parts of the whole network use different local schedules.

In this work, we assume that there is previous knowledge of the network topology. Therefore, we use a centralized scheduling method and distribute it to the nodes. We should note that this schedule has been created manually, but it is possible to generate one automatically using a controller, as can be seen in [8].

2.2. RPL

RPL [2] is a proactive distance-vector routing protocol for LLNs which maintains connectivity while minimizing energy consumption by limiting network traffic overhead. The nodes running the RPL routing protocol construct a DODAG, and the traffic is routed from these nodes to a border router, called the DODAG *Root*, that provides external connectivity. To achieve this, RPL initiates a routing instance to incorporate several nodes into one or many DODAGs. Then, depending on the instance's requirements, the nodes choose the best route for forwarding packets upwards (leaf to root). RPL supports two modes for performing downwards (root to leaves) routing:

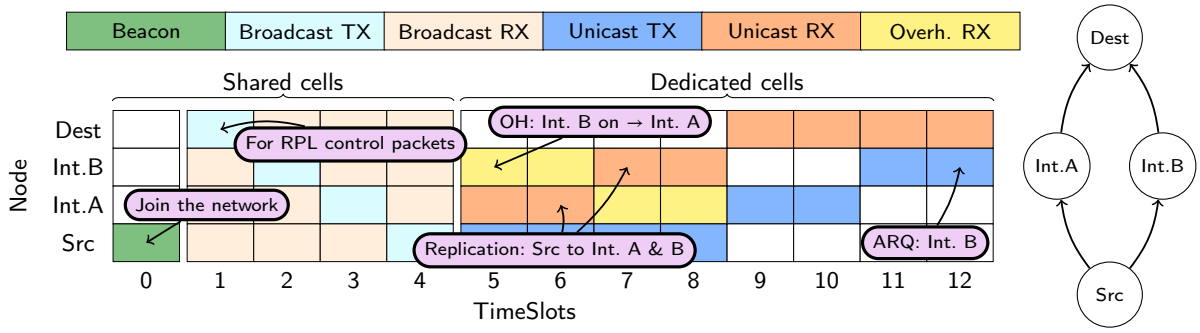


Figure 1: TSCH schedule example.

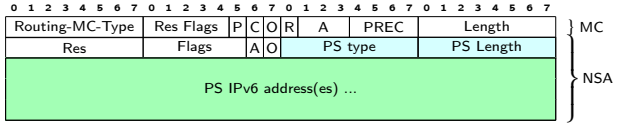


Figure 2: DAG Metric Container (MC) with nested Node State and Attribute (NSA) object.

- Storing mode: where intermediate nodes maintain state representing the routing information towards all their ancestors (direct children, children of children, etc.). This results in additional memory requirements for the nodes maintaining this information but higher network performance, especially for inter-DODAG traffic.
- Non-storing mode: where only the DODAG root maintains state representing the routing information towards the whole DODAG. This results in high memory requirements only on the root, but adds network overhead due to the use of source routing and less efficient intra-DODAG routing because all the traffic needs to go through the root.

RPL uses three types of control packets to build and maintain the DODAG.

- DODAG Information Object (DIO): Contains all the necessary information regarding a RPL instance, allowing a receiving node to connect. Given its nature, it is commonly transmitted as multicast and aims to grant the current DODAG information to a node so that it can make a parent selection.
- Destination Advertisement Object (DAO): Propagates reverse routing information and is transmitted every time a new preferred parent is selected. It is sent as unicast and delivered to the new parent node or the DODAG root, depending on the established RPL storing method (storing mode / non-storing mode).
- DODAG Informational Solicitation (DIS): Interrogates other nodes about the status of the network, soliciting DIO messages as a response.

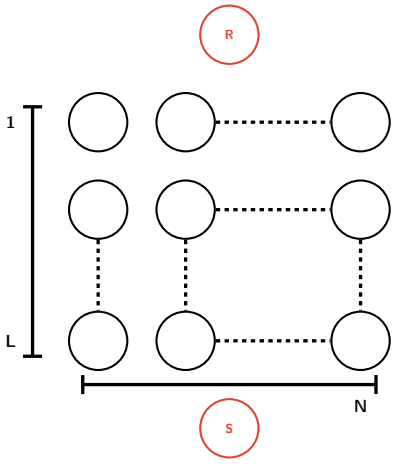


Figure 3: Grid-like network topology. Node S is the source node and node R is the DODAG root / destination node. The L intermediate layers each contain N nodes. Each node in each layer i is connected to all the nodes in the layer preceding it $(i-1)$.

To avoid loops in the DODAG, RPL performs distance-vector routing based on a rank hierarchy given by an Objective Function (OF). Out of a set of candidate parent nodes, the node with the lowest rank value is selected as the Preferred Parent (PP) and used for upwards routing. To calculate a rank, the OF can use different metrics [9], usually one of the two following:

- Hop Counting (HC): number of hops a packet must perform to reach its destination.
- Expected Transmission Count (ETX): Statistics aggregation that reveals a point-to-point link's quality. This metric reveals a route with the highest success probability.

3. Multi-path with the PAREO functions

The default implementation of RPL forwards packets from one node to another via a single-path. If one of the transmissions fails, the only option is to retry the transmission. Furthermore, transmission failure may occur not only

because of low link quality but also when one node in the path is unreachable. With the default single-path RPL, such node can be routed around, given time, with a local or global DODAG repair process [2]. However, in the time taken to detect this problem and repair the DODAG, packets will be delayed, and if the queues fill up, lost.

Multi-path routing addresses both cases thanks to the functions of PAREO [5], which are used to improve network reliability and energy efficiency. Additionally, they can work almost independently from each other. The individual functions that make up PAREO are described in the next sections.

3.1. Automatic Repeat reQuest (ARQ)

This function attempts to detect failed packet transmissions and performs a retransmission when that happens. In order for ARQ to function, some mechanism for detecting failed transmissions is needed. In this work we use the support provided by the MAC layer, TSCH. In TSCH, each unicast transmission is by default configured to require an acknowledgement, thus the lack of its reception can be interpreted as a failed transmission.

Using ARQ, given a link with a one-shot Packet Delivery Ratio (PDR) of p and up to r ARQ transmission attempts, the PDR with ARQ is $1-(1-p)^r$.

The disadvantage is that each attempt requires an additional cell, entailing an increase in delay, jitter, and energy consumption.

3.2. Packet Replication and Elimination (PRE)

Multi-path assumes using several routes to forward packet copies, i.e. *Replication*. As shown in [10], this increases the number of opportunities to successfully deliver the data packet to the destination. The primary parent used to forward data is called the PP while the rest of the used parents are the APs. In this work, all the multi-path algorithms use a single AP (i.e., two forwarders).

Additionally, it is possible that via different paths a node receives multiple copies of the same packet. If the node treats the copies as independent packets and also replicates each copy independently, then flooding can occur. To control this, PRE discards duplicated packets, i.e., *Elimination*. This is done by tagging each sent packet with a unique identifier at the source and keeping a recently seen identifier list on each node. If a received packet's unique identifier is in the list on a forwarding node, it is not forwarded but dropped.

3.3. Overhearing

In the context of using multiple parents, a device can also take advantage of the broadcast nature of the wireless medium and listen for transmissions intended for nodes different than itself. Indeed, by employing Overhearing [11], packet replication is performed more efficiently. To do so, we configure the nodes with an overhearing reception cell in their schedule. This allows keeping the nodes in listening mode when one of their neighbors is scheduled to receive a packet transmission. Note that the overhearing nodes do not send an acknowledgement when receiving a data packet. Only the unicast target destination node acknowledges the transmitter to avoid acknowledgment (ACK) collisions. The overhearing function is performed for both PP and AP, i.e., when the transmission is intended for the PP, the AP overhears, and when the transmission is intended for the AP, then the PP overhears.

4. Common Ancestor (CA) Algorithms

In this section, we present parent selection algorithms for multi-path routing based on a braided pattern with two routing paths. In the main path, nodes forward packets via the PP while in the alternative path nodes use another node called the AP. The way the AP nodes are selected greatly impacts the number of nodes that will be part of the forwarding path, and thus the energy consumption (the more the nodes involved, the higher the energy consumption). So it is important to define rules that allow constraining a transmission within a restricted number of nodes along a narrow path in the network. The CA approach [12] selects an AP if there is at least one potential parent in common between the parent sets of the PP and the AP. The objective behind this principle is to select an AP that is related to the PP in order to avoid (or reduce) the potential flooding and concentrate the transmission efforts towards a single direction. However, employing strict rules to select an AP may also reduce the probability of finding one, thus also impacting reliability. Given this trade-off between reliability and power consumption, three algorithms were defined, which vary in their flexibility when choosing an AP.

4.1. Overview

To be able to perform both a theoretical and an experimental analysis of the CA algorithm, we established a regular grid-like topology of $L \times N$, as shown in Figure 3, where N is the number of nodes per layer, L the number of layers between the source node (node S) and the DODAG root (node R), and $\mathcal{L}(i)$ is the set of nodes in the i^{th} layer. More specifically, $\mathcal{L}(0) = \{R\}$, $\mathcal{L}(L+1) = \{S\}$, and for the rest of the layers $i \in \{1, 2, \cdots, L\}$, $|\mathcal{L}(i)| = N$. Each node in each layer will have as its Parent Set (PS) the nodes located in the lower layer, i.e., $i > 0$ and $\forall k \in \mathcal{L}(i)$. $PS(k) = \mathcal{L}(i-1)$. Finally, the nodes report a subset of their parent set in the PS extension of the Node State and Attribute (NSA) object in the Metric Container (MC) of DIO messages (referred to as the PS_{MC} parent set from here on). This PS_{MC} extension contains a fixed number of addresses M, with $1 \leq M \leq N$.

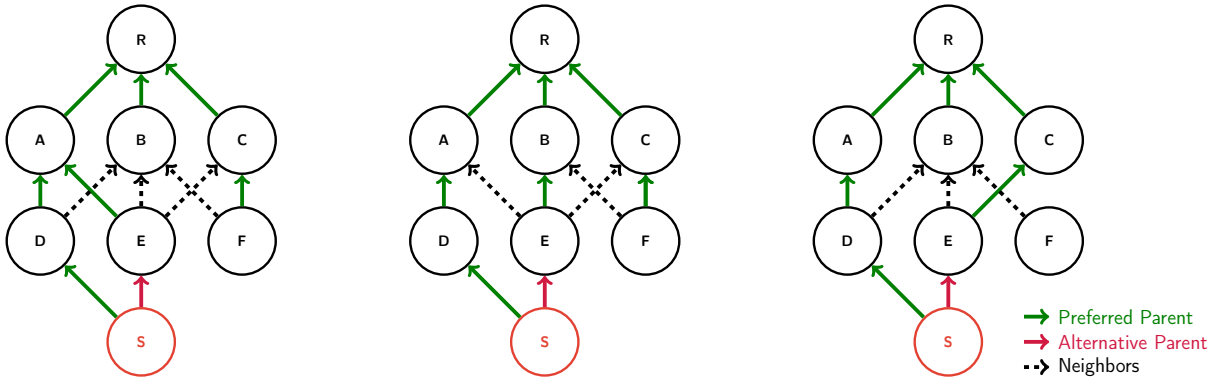
The probability of being able to find an AP depends on the AP selection algorithm used and the values of M and N. We will now investigate this probability since it significantly impacts the ability to implement braided multi-path routing.

4.1.1. Probability of obtaining an AP

The probability of obtaining an AP is given in Equation (2).

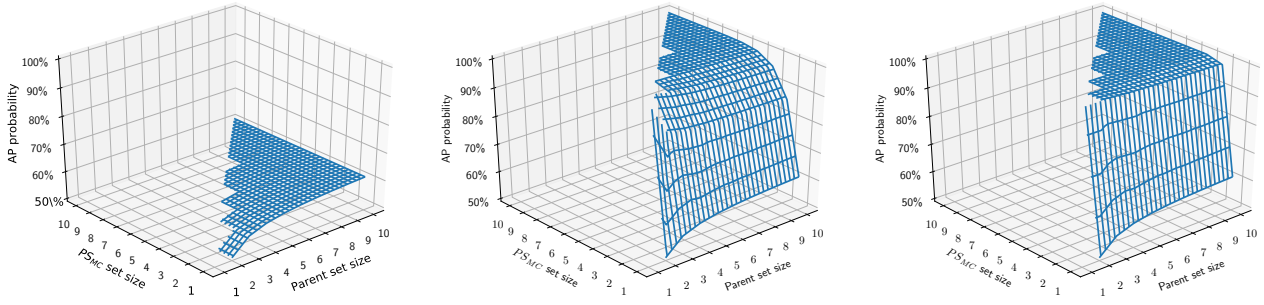
$$P(\exists AP) = 1 - (1 - P(\exists CA))^{N-1} \tag{2}$$

where $P(\exists AP)$ is the probability of having at least one AP and $P(\exists CA)$ is the probability of having at least one CA.



(a) Strict CA. (b) Medium CA. (c) Soft CA.

Figure 4: Algorithms for selecting an AP by using a Common Ancestor.



(a) Strict CA probability (b) Mediun CA probability (c) Soft CA probability

Figure 5: Probability of finding an AP through common ancestors parametrized over the PS set size N and the PS_{MC} set size M.

Different $P(\exists CA)$ are produced depending on the selection algorithm, and as a result also different $P(\exists AP)$.

4.2. CA Algorithms

Three different AP selection algorithms are presented in [12], with different trade-offs between the probability of selecting an AP and the level of flooding mitigation. In Figure 5, the probability of selecting an AP is shown for each of these algorithms, depending on the values of M and N.

4.2.1. Strict CA

Strict CA selects a candidate parent node c as an AP if the candidate node's PP is the same as the PP of the PP (the preferred grandparent) of the current node k, i.e., $PP(PP(k)) = PP(c)$. The probability of finding a CA is given in Equation (3).

$$P(\exists CA) = \frac{1}{N} \tag{3}$$

Given this, the probability of finding an AP is calculated in Equation (4).

$$P(\exists AP) = 1 - \left(\frac{N-1}{N}\right)^{N-1} \tag{4}$$

It is represented in Figure 4a with nodes A, D, E, and S. Since D and E have the same PP, which is A, then E is a candidate node to be selected as AP. Given the above and since E belongs to the PS of S, then E is selected as AP of S.

4.2.2. Medium CA

Medium CA selects a candidate parent node c as an AP if the candidate node's PS contains the PP of the PP (the preferred grandparent) of the current node k, i.e., $PP(PP(k)) \in PS(c)$. The probability of finding a CA is given by Equation (5).

$$P(\exists CA) = \frac{M}{N} \tag{5}$$

Using Equation (2), we then obtain the probability to get an AP, shown in Equation (6).

$$P(\exists AP) = 1 - \left(\frac{N-M}{N}\right)^{N-1} \tag{6}$$

The representation of its selection can be found in Figure 4b , where none of the nodes have a common PP except for those connecting to the root. However, since E has A, B, and C within its PS and D has A as PP, E is selected as AP.

4.2.3. Soft CA

Soft CA selects a candidate parent node c as an AP if the candidate node's PS has any common node with the PS of the PP (the preferred grandparent set) of the current node k, i.e., $PS(PP(k)) \cap PS(c) \neq \emptyset$. In Equation (7) the probability $P(\exists CA)$ of obtaining a CA depends on the probability of having a common node in the two PS_{MC} parent sets via the PP and via the candidate AP.

$$P(\nexists CA) = \frac{\binom{N}{M} \cdot \binom{N-M}{M}}{\binom{N}{M} \cdot \binom{N}{M}} = \frac{\binom{N-M}{M}}{\binom{N}{M}} \Rightarrow$$
$$P(\exists CA) = 1 - \frac{\binom{N-M}{M}}{\binom{N}{M}} \tag{7}$$

Finally, using Equation (2), the probability of finding an AP is shown in Equation (8).

$$P(\exists AP) = 1 - \left[\frac{\binom{N-M}{M}}{\binom{N}{M}}\right]^{N-1} \tag{8}$$

Its selection methodology is represented in Figure 4c, where E is selected as AP of S since its PP D has B as a common node with E. In case that multiple nodes are possible candidates to be selected as AP, then the one with the lowest RANK will be selected.

4.3. Common Ancestor Trade-offs

As it can be seen in Figure 4, in each algorithm, the PS of the PP of a node directly affects which nodes are eligible as APs. Even with Soft CA, the least restrictive selection algorithm, at least one node from the PS of the candidate AP has to belong to the PS of the PP of the node.

Indeed, if we analyze the three CA algorithms, the Strict CA selection performance, in terms of reliability, should be lower than the other two CA algorithms, since the probability of obtaining an AP is subject to share a common PP. As the probability is lower, fewer nodes are selected, and thus lower reliability is achieved, but less energy consumption as well.

A Soft CA selection, on the other hand, improves reliability since its AP selection works by matching any node between the PS of the PP with the PS of a possible AP. Therefore, the number of nodes employed for a transmission increases (and can easily lead to flooding), which in turn leads to higher energy consumption.

Using the notation defined in Section 4.1, let us consider a 3-layer topology between the source and the sink, where each layer is composed of 3 nodes, i.e., $L = 3$ and $N = 3$. If we make a transmission by making a Strict CA selection, we expect to get at most 6 relay nodes, i.e., the PP and AP in each layer. However, as seen in Figure 6, we can observe more selected forwarders since an AP does not necessarily have the same AP as the PP.

The Strict CA algorithm considers a node as a possible AP if both this node and the PP share the same PP. It is clearly possible that several nodes may fulfill this criteria, so another criterion might be used to select the AP, such as the candidate with the best link quality. However the choice is made, the selected node may select a different AP than the PP's AP, which will break the braided pattern and extend the number of nodes selected in each layer.

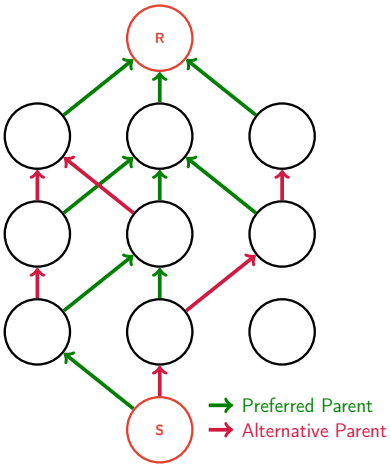


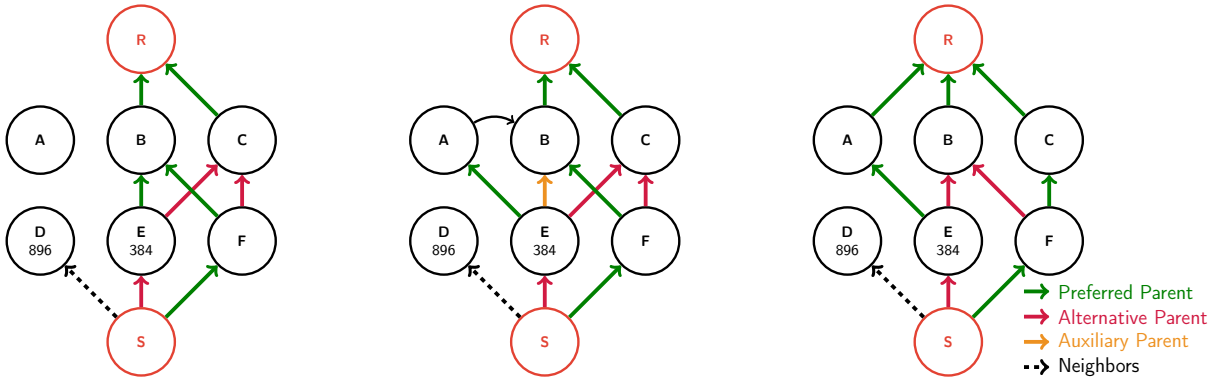
Figure 6: Strict CA problem statement.

Figure 6 demonstrates this behavior where it can be observed that all the selected parents belong to the algorithm condition, but at the same time, the selection is dispersed throughout the DODAG. It should be mentioned that each node does not necessarily have the same knowledge of its neighbors since a desynchronization may occur, e.g., the loss of a DIO packet. Also, since the parent selection depends on the parent's rank, there is a possibility of not obtaining an AP. Therefore, the packet will replicate uncontrollably, increasing the network overhead.

4.4. Implementation Aspects

The CA algorithms are implemented as custom RPL OFs, which extend The Minimum Rank with Hysteresis Objective Function (MRHOF) [13]. The same behavior as MRHOF is used as far as rank calculation and PP selection are concerned, but additional filtering of the candidate parents in a node's PS is performed for the AP.

As we show in [14], we have added the information required for this filtering, called the PS_{MC}, within DIO control packets. More specifically, the PS_{MC} is an additional field in the NSA object, which in turn is nested in the MC option of DIO control packets, as shown in Figure 2. The PS_{MC} field is stored as a list of IPv6 addresses, one for each potential parent in the PS, ordered by rank from most to least preferred (lowest to highest rank). Since IPv6 addresses require 16 bytes of storage, there is a limit to how many can be efficiently stored (thus $M \neq N$) and compression is generally required to send more than 2 addresses within the 127 byte frame limit of IEEE 802.15.4-2015. In our implementation, we only send the lower-order byte of this IPv6 address, following a compression method similar to the one used for the Source Routing Header (SRH) in the 6LoWPAN Routing



(a) Optimal scenario. (b) Parent change transition scenario. (c) Unsuited scenario.

Figure 7: ODeSe functions for parent selection.

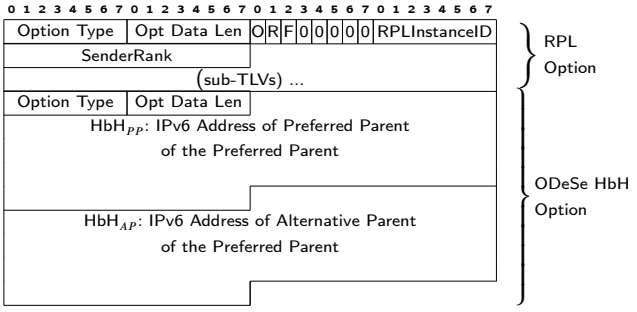


Figure 8: Hop-by-Hop option fields (RPL option and ODeSe field).

Header (6LoRH) [15].

5. ODeSe

While the Strict, Medium, and Soft CA algorithms perform well, the highest transmission reliability is achieved with Soft CA but at the cost of higher energy consumption due to higher flooding. In this section, we present ODeSe that aims to maintain the same level of network reliability while maintaining minimal transmission propagation between its parts. The main difference between the CA algorithms and the ODeSe is that in ODeSe each node decides not only its own forwarding parents (PP and AP) but also the PP and AP to be used in the next hop towards the destination. In other words, each node has a two-hop forwarding control. As it is shown in the performance evaluation, this results in tighter control of flooding that can be translated into lower energy consumption.

Given the above, ODeSe maintains as a selection policy the assignment of the same relay parents to the nodes that are at the same distance. However, ODeSe requires that these nodes point to the same PP. Otherwise, it loses the multi-path route direction.

Let us consider three different use cases. The first, illustrated in Figure 7a , represents the ideal ODeSe case. Here we can see that both E and F point to B as PP, allowing them

Algorithm 1: ODeSe.

Input: HbH$_{PP}$: The stored Preferred Parent within the Hop-by-Hop IPv6 extension header,
HbH$_{AP}$: The stored Alternative Parent within the Hop-by-Hop IPv6 extension header,
k: The current node.

Output: Returns the parent set used for the next data packet forwarding (PP, AP).

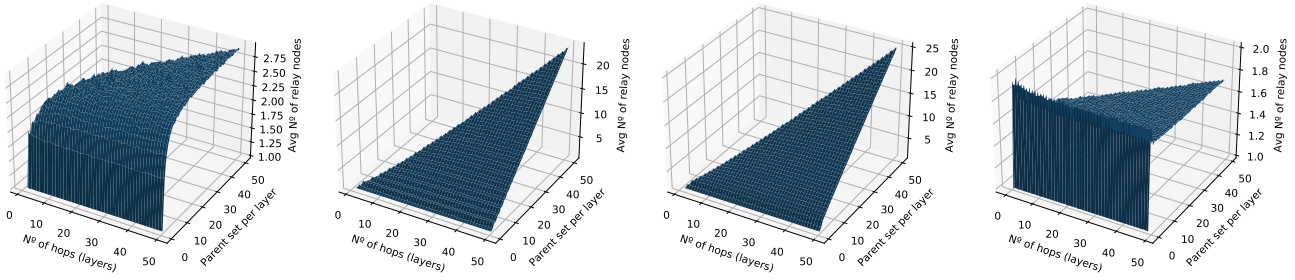
```
1  Function ODeSe(HbH_PP, HbH_AP, k)
2      PP_old ← PP(k) ;                          // Save current PP
3      if IS_VALID_PP(k, HbH_PP) then
4          PP(k) ← HbH_PP ;                      // Replace current PP
5      end if
6      if IS_VALID_AP_STRICT(k, HbH_AP) then
7          AP(k) ← HbH_AP ;                      // Use HbH_AP as AP
8      else
9          AP(k) ← NULL ;                        // Clear the AP before CA call
10         AP(k) ← CA(k, IS_VALID_AP_STRICT) ;   // AP Strict
11         if AP(k) = NULL then
12             AP(k) ← CA(k, IS_VALID_AP_MEDIUM)
13             if AP(k) = NULL then
14                 AP(k) ← CA(k, IS_VALID_AP_SOFT)
15             end if
16         end if
17     end if
18     PP_aux ← PP(k) ;                          // Save the PP to use for this packet
19     PP(k) ← PP_old ;                          // Restore old PP for other packets
20     return (PP_aux, AP(k))
21 end
```

where PP(k) and AP(k) correspondingly return the PP and AP of node k.

to choose C as AP.

When this situation does not occur, ODeSe has mechanisms that manipulate parent sets to adapt them to its requirements without affecting RPL. This is shown in Figure 7b, where an auxiliary link is generated to point to a suitable parent. More precisely, since E has A as PP, and F has B as PP, E must temporarily change its PP to B since it satisfies the requirements of the algorithm. This is because B is the PP of F, and F is the PP of S.

Finally, if a given scenario cannot be adapted to ODeSe, it will choose to make a selection using Soft CA (see Figure 7c).



(a) Strict CA. (b) Medium CA. (c) Soft CA. (d) ODeSe.

Figure 9:

5.1. Selection method

For every hop, a node k selects its PP and AP using the Strict CA algorithm. Additionally, in the forwarded replicas, the node stores the addresses of the PP and the AP of its PP, i.e., $\text{HbH}_{PP} = \text{PP}(\text{PP}(k))$, $\text{HbH}_{AP} = \text{AP}(\text{PP}(k))$. More specifically, these two addresses are stored within the Hop-by-Hop option header of the data packet, to be delivered to the next-hop node, see Figure 8.

As is shown in Algorithm 1, once received at the next hop node(s) $k' \in \{PP(k), AP(k)\}$ (i.e., both the PP and AP of k), the following checks are made. The next-hop node k' checks if the transported PP address HbH_{PP} is indeed a valid PP for it, and if yes, it then uses HbH_{PP} as the new PP for this specific packet, see Figure 7a and Figure 7b. The original PP is restored after finishing the forwarding operation for this packet. The validity check is required because information at node k might be "stale" due to not receiving up-to-date parent set information from k' via a DIO packet. If the HbH_{PP} is not a valid PP, then the default PP of node k' is used, see Figure 7c.

For the AP, node k' checks if the stored within the Hop-by-Hop option field HbH_{AP} node is also a valid AP using Strict CA. If it is, then k' uses HbH_{AP} as the AP (see Figure 7a). If it is not, then k' will try to obtain a new AP using first Strict CA, and if that fails, falls back to Medium CA and then Soft CA.

When k' has selected the AP and PP for the next hop, it does what node k at the previous hop did before it. It stores and replaces the addresses of the PP and the AP of its PP, i.e., $\text{HbH}_{PP} = \text{PP}(\text{PP}(k'))$, $\text{HbH}_{AP} = \text{AP}(\text{PP}(k'))$ in the forwarded replicas. Therefore, the entire process repeats until the data packet is delivered to the target node.

5.2. Multi-path algorithms behavior

To evaluate ODeSe against the three CA selection algorithms, we used the Monte Carlo method [16] using the grid topology explained in Section 4.1. This methodology consisted in modeling a DODAG construction using multi-path, where each node has a PP and a possible AP. The goal is to evaluate the average number of nodes used per hop "layer" during transmission between the source and the root node. To do so, we use different scenarios with hop distances L between 0 and 50 hops, and node sets per layer of size N between 0 and 50 nodes. The process of selecting a PP consists of choosing a random node from the upper routing layer L, based on the "distance" of a given node. Here, each node selected as a PP is provided with a probabilistic weight to increase the possibility of being selected again by other nodes.

Regarding the AP selection process, the principles of the CA and ODeSe algorithms are employed. Here, the selection method takes into consideration only a random half set of nodes contained in the upper hop layer L, based on the distance of a given node.

The results of this model are shown in Figure 9, where the x-axis indicates the distance L between the source and the root, the z-axis indicates the number of present nodes N per hop, and the y-axis indicates the average number of relay nodes per stage $L \times N$.

Figure 9a shows that the Strict CA algorithm increases the average number of relay nodes per layer until it normalizes in approximately three nodes.

Figure 9b and Figure 9c, on the other hand, have a linear upward increase. Due to the topology used, the differentiation between these two algorithms is quite small, being Medium CA a bit smaller than Soft CA in terms of the average number of relay nodes. Finally, ODeSe (Figure 9d), shows a horizontal linear average that does not exceed two nodes per layer yet with values close to those.

6. Simulation Setup

6.1. Simulation Parameters and Topology

We have employed the Contiki 3.0[1] operating system [17] and the COOJA simulator to implement the CA and ODeSe algorithms with a directed graph radio medium. Contiki comes with the default RPL protocol implemented, which is what is used for evaluating the single-path routing.

The network topology used for the evaluation is the one shown in Figure 3 with $L = 5$ and $N = 6$. The radio links in this topology come with 50% link quality on average, and every node (except the root and its direct children) has 6 potential parents to forward data. This link quality value has

[1] https://github.com/ariskou/contiki/tree/draft-ietf-roll-nsa-extension

been chosen following the Dust Networks[2] definition of a healthy network where each device should have at least a 50% link quality with its neighbors and at least 3 potential parents. Therefore, we evaluate three different cases with increasing average link quality and in all cases using a uniform random distribution for selecting the link quality values. Furthermore, the size of the PS_{MC} has been set to 3 to correspond to a more realistic case where finding a potential alternative parent even by employing Soft CA is not guaranteed.

At the MAC layer, we use TSCH with a timeslot length of $10ms$ and a total slotframe size of 297 timeslots, i.e., $2970ms$. The purpose of this is to provide the appropriate number of slots to locate 1 Retransmission (RTX) for each individual transmission and to locate all overhearing events as well.

A centralized TSCH scheduling was used as described in Section 2.1.3, where the slotframe takes its form based on Figure 1. For application traffic, in each simulation we send 250 UDP packets from a unique source node S to the root node R, with no fragmentation. The packets are sent once every 15 seconds so that at any one point in time there is only one packet being forwarded through the network.

For the routing algorithms, we have evaluated ODeSe against the three CA multi-path variants as well as against the default RPL single-path algorithm with different RTX count limits.

More specifically, the routing algorithms evaluated are:

1. Single-path (SP) with:
   (a) 0 RTX, which means no retransmission when an ACK is not received.
   (b) maximum 1 RTX upon failure.
   (c) maximum 3 RTX upon failure.
   (d) maximum 7 RTX upon failure.
2. Multi-path Strict CA with 1 RTX maximum.
3. Multi-path Medium CA with 1 RTX maximum.
4. Multi-path Soft CA with 1 RTX maximum.
5. *Multi-path On Demand Selection CA with 1 RTX maximum*.

For each of the above routing algorithms, we executed 20 simulation repetitions with 20 different Pseudo-Random Number Generator (PRNG) seeds, to increase the statistical reliability of our results. In total, therefore, 5000 data packets have been sent for each routing algorithm, and a total of 480 simulations have been executed.

In Table 1, the detailed simulation parameters are listed.

Table 1
Simulation parameters.

Simulation		
Replications/PRNG seeds		20
Duration		until 250 pkts sent
Data traffic		$1pkt/15sec$
Routing		RPL
Parent set size (N)		6
PS_{MC} size (M)		3
UDP payload size		16 bytes
TSCH	Single-path	Multi-path
Scheduling	Centralized	Centralized
EB period	4 sec	4 sec
Timeslot length	10 ms	10 ms
Slotframe length	297 Timeslots	297 Timeslots
Nº of channels	1	1
Nº of RTX	0 RTX, 1 RTX 3 RTX, 7 RTX	1 RTX
Topology		
Topology		Multi-hop
Nº of nodes		32
Nº of layers (L)		5
Nº of sources		1
Link quality		Uniform distribution (U(a, b)): U(40%, 60%), $\mu = 50\%$ U(50%, 70%), $\mu = 60\%$ U(50%, 90%), $\mu = 70\%$

6.2. IEEE Std 802.15.4-2015 TSCH Schedule

This work evaluates routing algorithms within RPL, which is MAC agnostic. For the MAC layer, we have chosen to use TSCH for its scheduled nature in order to reliably evaluate energy consumption, delay, and jitter.

For UDP data transmissions, the experimental scenario that we investigate cannot have transmission collisions. Thus, only 1 TSCH channel is used for simplicity. Furthermore, we wanted to allow the routing protocols to select at each node any of the nodes in the higher layer as a PP or an AP. Therefore, we created a schedule which for each node has transmission cells to all the possible parents. Additionally, for each possible parent, there are two available cells to support 1 RTX within the same slotframe.

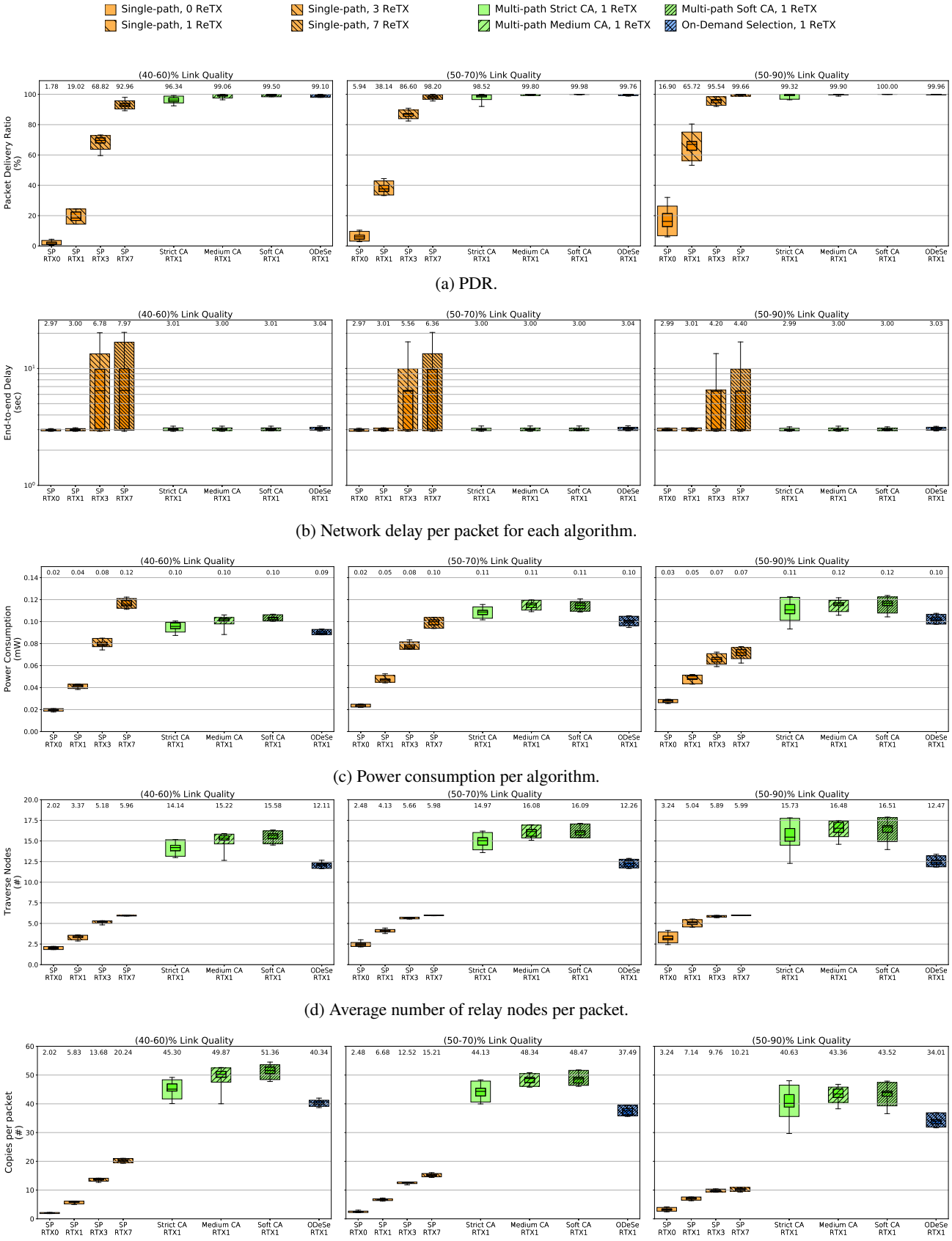
Finally, overhearing is not supported in TSCH, so it has been implemented as an extension in Contiki. When any node is transmitting to one of the parents, overhearing cells are placed in the rest of the potential parents. They take the form of shared receive cells (i.e., listening for packets from any source) with the normal MAC filtering disabled (i.e., unicast packets received are not checked if the receiver is the intended destination). An example showing a simplified version of the schedule is shown in Figure 1.

7. Performance evaluation

7.1. Packet Delivery Ratio (PDR)

During single-path transmission, the probability of success will depend on the link quality and the number of retransmissions per packet lost. As already mentioned in this investigation, only four attempts (SP-3RTX) are required to transmit a single-hop transmission with $\text{PDR}_{\text{Single Hop}} = 93.75\%$ using a 50% link quality on average (i.e., as in the U(40%, 60%) case). Since each hop transmission is independent, the PDR between the source to the sink is given by the multiplication of the probability of each hop, i.e., $\text{PDR} = \text{PDR}^{L}_{\text{Single Hop}}$. Assuming that the total number of hops is six, then PDR = 68.82% (U(40%, 60%), SP-3RTX)

[2] SmartMesh IP Application Notes, Linear Technology Corp. 2012-2016.



(a) PDR.

(b) Network delay per packet for each algorithm.

(c) Power consumption per algorithm.

(d) Average number of relay nodes per packet.

(e) Average number of copies of a packet per algorithm.

Figure 10: Performance evaluation

as shown in the approximate value of Figure 10a. Multi-path, on the contrary, uses two parents per hop transmission. The probability that both receive the packet with also four opportunities is 87.89%. Therefore, if two nodes transmit to the same parent (CA and ODeSe), the probability of a hop success is 99.60%, since eight attempts are used (four attempts per child). Consequently, reliability improves. In evaluating different link quality ranges, it can be seen in Figure 10a that as the average link quality increases ($\mu = \{50\%, 60\%, 70\%\}$) also the PDR increases for all algorithms, with ODeSe achieving a PDR between 99.1% and 99.96%.

In all cases ODeSe outperforms SP-RTX7 and is placed consistently between Medium CA-RTX1 and Soft CA-RTX1.

7.2. Delay

Due to the employed centralized scheduling, the slotframe size of the schedule severely impacts delay performance. This requires to obtain all possible transmission cases between the source and the DODAG root. Considering the topology, see Figure 3 with $L = 5$ and $N = 6$, the total number of nodes between the source and the sink is 30. If we consider that each node within the topology has 6 possible parents, excluding the DODAG root and its direct children, then the total number of upward links is 156. More specifically, if T is the total number of links

$$\begin{aligned} T_{L+1} &= N \text{ (source layer)} \\ T_{L\cdots 2} &= N^2 \text{ (each intermediate layer)} \\ T_1 &= N \text{ (layer of direct children of the root)} \\ T &= L_{L+1} + (L-1)L_{L\cdots 2} + L_1 \\ &= (L-1)N^2 + 2N \end{aligned}$$

Finally, if we consider that there is one retransmission per slotframe (i.e., two transmissions) and that each timeslot lasts 10 ms, then the maximum delay per sloframe is 3120 ms, i.e., if we consider MD as maximum delay per sloframe, then $\text{MD} = \text{TL} \times 2 \times 10\ ms$.

Figure 10b shows this analysis, with single-path algorithms which perform more than one retransmission (RTX > 1) affected since they require more slotframes to transmit. In contrast, all the multi-path algorithms, i.e., ODeSe and Strict, Medium, and Soft CA, only use one slotframe per end-to-end transmission.

7.3. Energy Consumption & Traversed Nodes

The simulator used, COOJA, can provide energy consumption information in terms of duty cycle for each radio state and for each node. We have used that duty cycle information to extrapolate the average power consumption per node. To do so, we used the radio power consumption values of the Zolertia Z1 mote with the CC2420 radio transceiver module. The values taken from its specification sheet ($P_{TX} = 52.2mW@3V$, $P_{RX} = 56.4mW@3V$) are used as factors for each radio state's duty cycle.

As it can be observed from Figure 10c, single-path comes with better performance than the multi-path algorithms, with single-path reducing the energy consumption as the links become more reliable and as a result fewer retransmissions being used. The multi-path algorithms in general do not rely as much on retransmissions, and therefore have less room for energy consumption improvement due to this factor. This is mainly due to the number of nodes used to forward a single data packet and due to the overhearing operation that requires an extra timeslot. Among the multi-path algorithms, Soft CA comes with the highest energy consumption due it's production of flooding, while ODeSe consumes the least energy, since it concentrates its routes without dispersing them over the DODAG, see Figure 10d.

Note that the energy consumption calculation comprises the cost sending both data and control packets.

7.4. Number of Copies per Single Packet

The total number of copies per data packet consists of the number of times a data packet was forwarded at the relay nodes or at the source node. As we have seen in Section 4.3, CA algorithms do not regulate the selected APs if they can satisfy the selection policies of each algorithm. However, since ODeSe aims for all nodes of the same hop distance to forward their links to the same PP and AP, this algorithm is expected to have a restricted number of relay nodes, with 2 parents per node, as well as a limited number of copies of a packet.

As can be seen in Figure 10e, the total number of copies per original data packet in ODeSe decreases by 19.794% compared to the average of CA algorithms, demonstrating that ODeSe's selection policy minimizes the traffic load in multi-path routing.

7.5. Discussion

As can be seen in the results of Figure 10, in general, the use of multiple routes increases the reliability of transmission with low delay but also consumes a greater amount of energy, since a greater number of nodes are used to send a packet. On the other hand, the concentration of the alternative paths closer to the main path is necessary for higher energy efficiency.

To demonstrate this behavior, we made several evaluations using the Pareto principle. These graphs illustrate the contribution of each algorithm in terms of nodes used and power spent per transmission.

Figure 11a shows that ODeSe uses a lower number of copies to achieve a PDR comparable to other algorithms, reaching values close to 100%.

Figure 11b, on the other hand, shows that additionally, ODeSe consumes less energy than the CA algorithms.

In both cases, comparing the performance of ODeSe against the other CA algorithms, the CA algorithms use more nodes per transmission and, more energy. Additionally, not all CA algorithms have the same transmission reliability, e.g., Strict CA and Medium CA have worse performance than ODeSe. The highlighted Pareto frontier can help in choosing an appropriate algorithm depending on the required levels of power consumption and reliability.

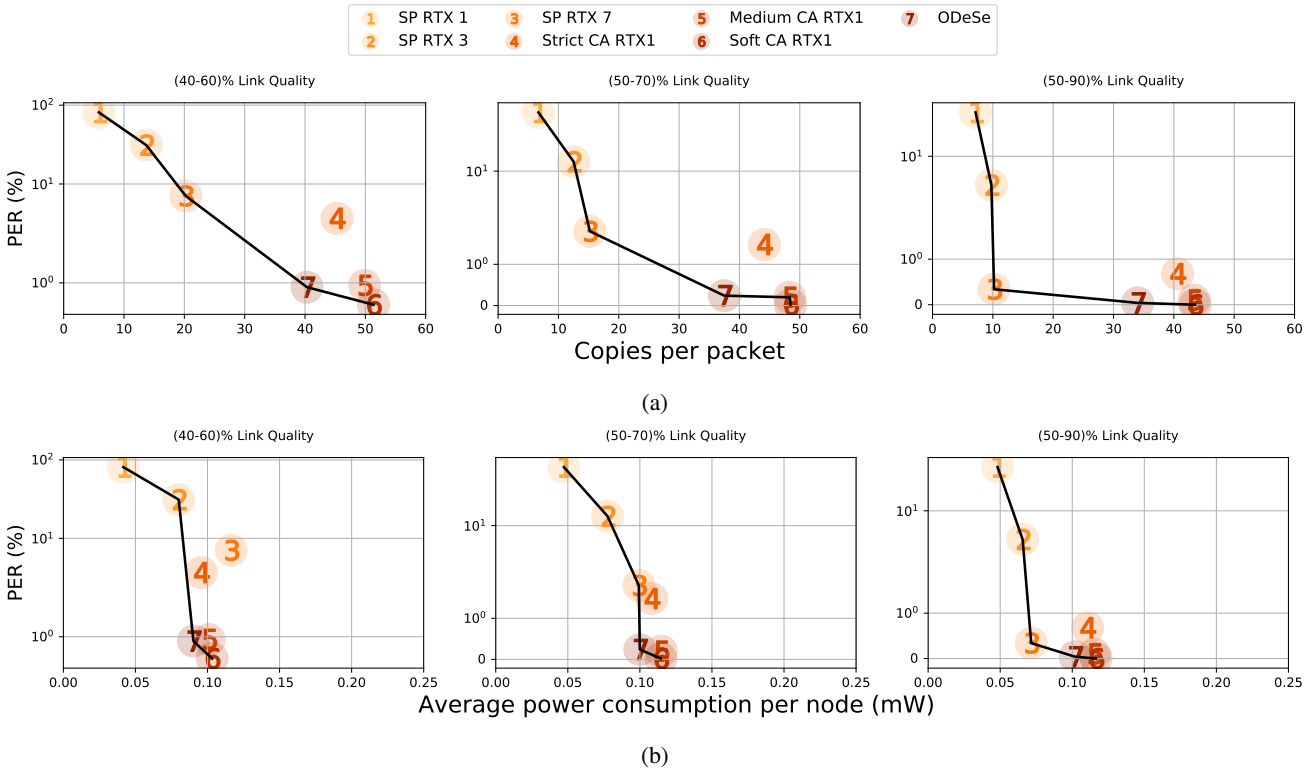


Figure 11: Relationship between (a) the average copies per packet transmitted and the end-to-end PER per algorithm, (b) the average power consumption per node and the end-to-end PER per algorithm. The line represents the Pareto frontier.

8. Related work

The authors in [18] compared three types of multi-path algorithms that guarantee a network reliability improvement. First, they defined a Disjoint pattern, which aims to propagate different alternative routes isolated from each other, that is, without a common node between them. The routes are generated from the source node and merged at the sink. Then, they established a triangular pattern, which allows the nodes of the main path to have APs based on a common ancestor. This pattern does not allow alternate nodes to select an AP. Finally, they established a braided pattern, whose goal is to select an AP if there is a common ancestor. Unlike the triangular pattern, this pattern allows the alternative nodes to also select an AP. The authors specify that algorithms with common ancestors improve the reliability of the network at the cost of overhead since there are more routes where the packets can go. The existing problem with the braided pattern is that it can spread across the topology since several common ancestors can exist, therefore, a controlled braided pattern is required.

In [19], the authors propose the use of multiple routes by selecting an AP using the braided pattern. This proposal aims to use the AP if the PP cannot be reached. Furthermore, they propose the use of a centralized TSCH scheduler that is constructed in such a way that the nodes can perform an end-to-end transmission from the source to the sink in a single slotframe. To take advantage of the isolation that exists between the PP and AP nodes, the authors propose to assign the RX of the nodes of the same layer in a shared cell and the TX of the nodes that will transmit them in dedicated cells. In this way, if there is a common receiver, the transmission from a PP or an AP will use the same timeslot. What is not taken into account is the existence of desynchronization in the reception of the packet. If we consider the transmission of a received data packet to be successful, an ACK is expected to verify its delivery. Since these packets can get lost, this can result in the AP being used for transmitting as well, thus generating collisions.

The authors in [20], [21] and [22] propose the use of multiple routes to reduce upstream traffic congestion in an RPL network. Lodhi et al. [20] propose the use of the M-RPL extension which has the purpose of detecting the traffic of the forwarding nodes by using the PDR based on a series of expected packets. As soon as a node's PDR exceeds the congestion threshold, it alerts its children to select an alternate parent and thus distribute traffic over two transmission routes instead of one (PP and AP). Iova et al. [21], on the other hand, defined an Expected Lifetime metric which represents the amount of traffic that the nodes is handling. Depending on the values provided by this metric, the nodes balance the amount of traffic to their neighbors so that each node in the network has similar traffic. Le et al. [22] propose three types of algorithms to select an AP in RPL: Energy Load Balancing (ELB), Fast Local Repair (FLR), and ELB and FLR together. ELB has an OF that uses hop-counting and residual energy metrics to select nodes with the highest energy

levels. In this way, the neighbor with the lowest rank will be selected as PP, while the set of APs will be considered as a backup. FLR, on the other hand, attempts to reduce the number of local repairs. This is achieved by searching for neighbors with the same rank as the rank of the PP, with the objective of using them as PP if the actual PP requires a local repair. Finally, the combination of the two algorithms results in a reduction in overhead and delay while a well-balanced network is maintained. In all three of these works, although the authors balance traffic to multiple parents, this does not guarantee that the network is reliable. This is due to the fact that the transmission opportunities are scattered throughout the topology, which can cause disjoint paths that in turn depend on re-transmissions due to their isolation.

LeapFrog Collaboration (LFC) is a RPL-based multi-path routing algorithm that maintains reliability by taking advantage of the properties of the wireless environment, and packet replication and elimination functions. It uses a braided pattern based on common ancestor strategy, where the preferred and the alternative parents of a node should have at least one common parent. Furthermore, to offer more receiving opportunities, LFC uses the overhearing function to keep the alternative node in reception mode in order to capture the transmitted packets when a node transmits to its preferred parent. These methods result in an algorithm which maintains high reliability at the expense of energy consumption [23] [3]. In contrast, the CA algorithms and ODeSe have been designed with the intent of more flexible parent selection to control flooding. As a result, Strict CA and ODeSe are more energy efficient.

9. Conclusion

In this work, we addressed the trade-off between reliability and energy consumption in the industrial IoT context, and more specifically when using IEEE Std 802.15.4-2015 TSCH as the MAC protocol and RPL as the routing protocol. Previous works on multi-path with CA achieve high reliability using multi-path techniques but at the cost of high energy consumption. We propose here the ODeSe algorithm which achieves very high reliability with lower energy consumption, thus providing better options in the reliability - energy consumption trade-off. More specifically, the results show that it is possible to attain the high reliability offered by the highest energy consuming CA algorithm (Soft CA) with energy consumption lower than the least energy consuming CA algorithm (Strict CA). Additionally, ODeSe offers a better trade-off than the highest reliability single-path algorithm evaluated (Single-path RPL with 7 RTX). Therefore, ODeSe offers a novel solution for high-reliability industrial wireless networks.

It may be possible to further improve performance, and in the next steps we will investigate the possibility of reducing the energy consumption of nodes by more intelligently choosing when nodes enable their radios to listen, by taking advantage of the information in the ODeSe control packets.

10. Acknowledgment

The Cooja simulations presented in this research were performed on the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). Additionally, this work was partially performed and supported under the TPI ANR-17-CE10-0007-01 project of the French National Research Agency (ANR).

References

[1] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), 2016.

[2] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and Alexander R. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550, 2012.

[3] G. Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, and N. Montavont. Leapfrog Collaboration: Toward Determinism and Predictability in Industrial-IoT applications. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Paris, France, 2017.

[4] P. Minet, I. Khoufi, and A. Laouiti. Increasing reliability of a TSCH network for the industry 4.0. In *Proceedings of the IEEE International Conference on Network Computing and Applications (NCA)*, Cambridge, USA, 2017.

[5] R.-A. Koutsiamanis, G. Z. Papadopoulos, T. Lagos Jenschke, P. Thubert, and N Montavont. Meet the PAREO Functions: Towards Reliable and Available Wireless Networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020.

[6] T. Watteyne, M. Palattella, and L. Grieco. Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. IETF RFC 7554, 2015.

[7] J. Rugamba, D. Mai, and M. Kim. Implementation of a Centralized Scheduling Algorithm for IEEE 802.15.4e TSCH. In *Proceedings of the International Conference on Intelligent Computing (ICIC)*, Nanchang, China, 2019.

[8] R.-A. Koutsiamanis, G. Z. Papadopoulos, B. Quoitin, and N. Montavont. A Centralized Controller for Reliable and Available Wireless Schedules in Industrial Networks. In *Proceedings of the International Conference on Mobility, Sensing and Networking (MSN)*, Tokyo, Japan, December 2020.

[9] J.P. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel. Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks. IETF RFC 6551, 2012.

[10] G. Z. Papadopoulos, R.-A. Koutsiamanis, and N. Montavont. Exploiting Packet Replication and Elimination in Complex Tracks in LLNs. draft-papadopoulos-paw-pre-reqs-01, 2019.

[11] E-N Huh G-W. Lee. Reliable data transfer using overhearing for implicit ACK. In *Proceedings of the ICROS-SICE international joint conference (ICCAS-SICE)*, Fukuoka, Japan, 2009.

[12] T. Lagos Jenschke, G. Z. Papadopoulos, R.-A. Koutsiamanis, and N. Montavont. Alternative Parent Selection for Multi-Path RPL Networks. In *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 2019.

[13] O. Gnawali and P. Levis. The Minimum Rank with Hysteresis Objective Function. IETF RFC 6719 , 2012.

[14] R.-A. Koutsiamanis, G. Z. Papadopoulos, N. Montavont, and P. Thubert. RPL DAG Metric Container Node State and Attribute object type extension, 2020.

[15] P. Thubert, C. Bormann, L. Toutain, and R. Cragie. IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header. IETF RFC 8138, 2017.

[16] W. D. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[17] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and

flexible operating system for tiny networked sensors. In *Proceedings of the IEEE International Conference on Local Computer Networks (LCN)*, Tampa, FL, USA, 2004.

[18] P. Minet, I. Khoufi, and A. Laouiti. Increasing reliability of a TSCH network for the industry 4.0. In *Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, 2017.

[19] E. Mozaffari, M. Nassiria, and F. Theoleyre. Multipath aware scheduling for high reliability and fault tolerance in low power industrial networks. *Network and Computer Applications*, 142:25–36, 2019.

[20] M. A. Lodhi, A. Rehman, M. M. Khan, and F. B. Hussain. Multiple path RPL for low power lossy networks. In *Proceedings of the IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, Bandung, Indonesia, 2015.

[21] O. Iova, F. Theoleyre, and T. Noel. Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Networks*, 29:45 – 62, 2015.

[22] Q. Le, T. Ngo-Quynh, and T. Magedanz. RPL-based multipath Routing Protocols for Internet of Things on Wireless Sensor Networks. In *Proceedings of the International Conference on Advanced Technologies for Communications (ATC)*, Hanoi, Vietnam, 2014.

[23] R.-A. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. Del Fiore, P. Thubert, and N. Montavont. From Best-Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks. *IEEE Transactions on Industrial Informatics*, 14:4468–4480, 2018.