



HAL
open science

SDN-based approach for adaptive reconfiguration of routing in IoT for smart-buildings

Liticia Djennadi, Gladys Diaz, Khaled Boussetta, Christophe Cerin

► **To cite this version:**

Liticia Djennadi, Gladys Diaz, Khaled Boussetta, Christophe Cerin. SDN-based approach for adaptive reconfiguration of routing in IoT for smart-buildings. HPSH 2024 - 25th IEEE International Conference on High Performance Switching and Routing, Jul 2024, Pisa, Italy. pp.137-142, 10.1109/HPSR62440.2024.10635953 . hal-04923488

HAL Id: hal-04923488

<https://hal.science/hal-04923488v1>

Submitted on 31 Jan 2025

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.



Distributed under a Creative Commons Attribution 4.0 International License

SDN-based approach for adaptive reconfiguration of routing in IoT for smart-buildings

Liticia Djennadi*, Gladys Diaz*, Khaled Boussetta* and Christophe Cerin†

*Laboratoire de Traitement et Transport de l'Information

†Laboratoire d'Informatique de Paris Nord

Institut Galilée, Université Sorbonne Paris Nord

99 Avenue Jean-Baptiste Clément, 93430 Villetaneuse, France

Abstract—Software Defined Networking (SDN) is a widely adopted approach for wireless network reconfiguration. However, its use in IoT context is still challenging. A fundamental issue is related to the expensive overhead of SDN signaling in an IoT architecture where the lowest level is composed of a multi-hops Wireless Sensor Network (WSN). This work aims to contribute on overcoming such issues. Precisely, we investigate the improvement of SDN efficiency in the case of dynamic reconfiguration of Low-Power and Lossy Networks (RPL) parameters. To conduct our study, we focus on μ SDN, a lightweight SDN architecture designed for IoT environments and developed under Contiki framework. Three key enhancements are introduced to μ SDN. First, a mechanism that regulates SDN signaling messages accordingly with the state of network topology. Second, a proactive path establishment to mitigate network access delay for scheduled traffic requests. Third, a dynamic configuration of RPL to adjust the energy consumption accordingly with two modes: a day mode where human activity in the smart building is intense, and a night/holiday mode where less traffic is generated by the sensors. Performances evaluation of our version, referred to as e- μ SDN (enhanced μ SDN), shows that our proposal significantly reduces the overhead of SDN signaling messages and the induced energy consumption while improving the Packet Delivery Ratio. We also show the ability of e- μ SDN to dynamically adjust RPL parameters following the day and night modes.

Index Terms—SDN, RPL, IoT, Smart-Building, Protocol re-configuration, Signaling overhead, and energy efficiency.

I. INTRODUCTION

Smart buildings are designed to provide sustainable and user-friendly environments by leveraging Internet of Things (IoT) technologies. A classical multi-tier IoT architecture consists of several layers: Device, Fog, and Cloud. Each layer is in charge of fulfilling specific functions. The lowest layer, composed of IoT devices, is in charge of collecting data from source nodes and transmitting the resulting data to the customers' terminals. In typical smart buildings use cases, data are collected from sensors, which are often fixed at the deployment phase. It is also possible to integrate mobile data sources, such as smartphones, to enrich data and sustain human-centric services. The nodes can be either static (e.g. actuators) or mobile (e.g. portable devices). The fog and cloud levels are in charge of data storage, processing, and the management and control of the infrastructure. The interconnection between levels (device-Fog, Fog-Cloud), is achieved through different network protocols and technologies.

In this work, we focus on the wireless communications between IoT devices and the fog nodes. Different access technologies can be used, enabling either the single-hop (e.g. 5G or LoRa) or the multi-hop (e.g. IEEE 802.11x or 802.15.4) mode. We investigate the latter category, which is still widely deployed in indoor environments. It is also the most challenging one; since a routing protocol is required to ensure the connectivity among devices and fog nodes.

Precisely, we consider Low-Power and Lossy Networks (RPL) [1], which is one of the most popular protocols defined by the IETF for multi-hops routing between wireless IoT devices. RPL relies on IPv6, a staple for IoT services. It has been specifically tailored for low-cost IoT devices that have limited processing, communication, and energy resources. In particular, those that are powered with batteries and rely on unlicensed radio access technologies (e.g. IEEE 802.15.4), which can experience packet losses resulting from interference and contention. In addition, RPL is widely implemented in plenty of IoT platforms (e.g. Contiki, RIOT, Renode, Raspbian). As it is still the most common scenario in current smart-building use-cases, we consider RPL for static nodes. The support of mobile nodes will be investigated in future works.

In practice, the configuration of RPL parameters is still set nowadays at the deployment stage. Several works [2] [3] have shown that RPL parameters can have a significant impact on energy consumption, packet losses, network lifetime, and various other performances. Nevertheless, static configuration of RPL parameters is not efficient in environments like smart buildings, where data flows and communication resources are strongly correlated to human activities. As a matter of fact, the traditional approach of configuring parameters during the network's initialization lacks the adaptability needed to cater to the temporal dynamics of IoT services.

The reconfiguration of network parameters is a long-standing concern in the field of networking. Software-Defined Networking (SDN) emerges as the most opportune solution due to its ability to decouple control and data planes, enabling dynamic control and programmability of network devices [4]. This architectural shift facilitates centralized management, real-time reconfiguration, and the crucial adaptability to cope with the diversity and evolving demands of IoT environments. However, the direct application of traditional SDN paradigms to lightweight embedded systems, such as Contiki, comes up

against difficulties inherent to the limited hardware capacities of IoT devices [5]. In particular, an accurate and updated view of SDN controllers is possible at the expense of a high number of signaling messages, representing a significant bandwidth overhead at the radio interface and a large amount of consumed energy. Different works have been proposed in the literature to tackle this challenge. μ SDN framework is one of the more remarkable proposals [6].

In this paper, we propose an enhanced version of μ SDN. Our proposal, so-called e- μ SDN, integrates three mechanisms. The first mechanism aims to reduce the overhead of SDN control messages. The second mechanism seeks to minimize the network access delay for periodic traffic. Finally, the last mechanism allows an adaptive configuration of RPL to accommodate with different data traffic intensities induced by cyclic human activities in smart buildings.

The remainder of the paper is organized as follows: Section II provides background on RPL and related works on SDN for IoT. Section III describes our proposal, the so-called *enhanced* μ SDN. Performance evaluation is described and discussed in section IV. Finally, conclusions are drawn in section V.

II. STATE OF THE ART

A. Background on RPL

RPL is a proactive routing protocol specifically designed for low-power networks with low data rates [1]. The network topology is represented by a directed acyclic routing tree called a Destination-Oriented Directed Acyclic Graph (DODAG). Generally, the root is a gateway node connected to the Internet. The selection of the optimal path to each node in the network is determined by an objective function, which could include different metrics such as the number of hops, link quality, or residual energy. To build and maintain the topology, RPL defines four types of messages :

- *DODAG Information Object (DIO)* is a multicast message sent by parent nodes to child ones to disseminate information needed to establish the topology.
- *DODAG Information Solicitation (DIS)* is sent by a node aiming to join the DODAG and requesting information about the network topology.
- *Destination Advertisement Object (DAO)* is sent by child nodes to their parents, announcing their availability to route packets to a particular destination.
- *Destination Advertisement Object Acknowledgement (DAO-ACK)* is sent back by parent nodes to confirm the reception of a DAO message.

The transmissions of those messages occur through intervals, dimensioned through some RPL parameters. However, DIO messages, which are sent in multicast, are the most common exchanges, inducing a significant energy consumption. To mitigate this impact, *Trickle* algorithm [7] is employed to reduce the number of DIO messages transmitted when the topology is consistent. The transmission interval of those messages denoted I , is established within a predefined range $[I_{min}, I_{max}]$. Here, I_{min} and I_{max} represent the minimum

and maximum waiting time before transmitting a DIO message. In general, I is initialized to I_{min} . A DIO transmission is triggered within the interval $[I/2, I]$ only when an inconsistent message has been received or if the number of received consistent messages within I is below a redundancy constant, noted K . If an inconsistent message has been received, I is reset to I_{min} , otherwise I is doubled until it reaches I_{max} .

B. Related works on SDN for IoT

SDN is a new paradigm that revolutionizes network management by separating the control plane from the data plane. This innovative approach externalizes the intelligence of the network to a software entity, called the *controller*.

SDN was initially applied to networks that often involve powerful network equipment. However, recent efforts have been made to extend the SDN to sustain the development of IoT services, such as for smart buildings. Among the few architectures that have been recently proposed, one can cite μ SDN [5], CORAL-SDN [8], SBR [9] and SDN-WISE [10].

μ SDN [5] is a lightweight SDN architecture specifically tailored to optimize the communications between the controller and IoT devices. It is compatible with IPv6 and uses RPL to provide communication paths.

CORAL-SDN [8] is another SDN alternative for IEEE 802.15.4, IPv6 and RPL-based IoT networks. CORAL-SDN defines a customized external controller. As indicated in [11] the use of an SDN external controller induces more energy consumption and additional latency when compared to an embedded controller, as in μ SDN.

SBR [9] is an innovative architecture that introduces an SDN architecture, utilizing the RPL routing protocol. However, unlike μ SDN, SBR Code is not publicly available.

SDN-WISE [10] extends the OpenFlow approach of SDN and introduces a customized routing protocol, rather than using the RPL standard, as in μ SDN. In [11], SDN-WISE and μ SDN are assessed. The study compares the performances of these two architectures in terms of energy consumption, average round-trip time between the controller and nodes, end-to-end latency, and packet delivery ratio. Performance evaluation was conducted using *Cooja* simulator. The results reveal that μ SDN outperforms SDN-WISE across all previously mentioned parameters. Consequently, we select μ SDN to investigate the adaptive reconfiguration of RPL parameters for IoT-based smart building services.

III. PROPOSAL: ENHANCED μ SDN

A. Problem statement

As shown in [5], μ SDN exhibits a slightly lower Packet Delivery Ratio (**PDR**) and introduces an additional energy cost, compared to RPL. Indeed, the SDN paradigm adds specific signaling overhead that can lead to a significant energy drain for IoT devices. Actually, the traffic generated by μ SDN primarily stems from three processes: *controller discovery*, *node state updates*, and *requests for instructions to the controller*. The former process is punctually performed at the initialization phase of a node seeking to join the topology.

Its associated overhead is thus marginal, compared to the other processes.

Indeed, the significant overhead results from *Node State Update (NSU)* messages, carrying information about nodes' states. Their purpose is to provide the SDN controller with an up-to-date overview of the network topology and allow it to accordingly determine optimal flow paths. Those messages are generated by each node in the network and are periodically sent to the SDN controller node. In the most recent implementation of μ SDN under Contiki-3.0, the periodicity of NSU messages is statically set to 1 minute. Nevertheless, in [5], the performance evaluation study was conducted with an NSU periodicity fixed to 3 minutes.

In the next subsection, we will introduce an *enhanced* version of μ SDN, that aims to reduce the overhead of NSU messages. Our proposal, referred to as *e- μ SDN*, adds also a proactive path establishment mechanism, which seeks to minimize network access delays for periodic traffics. Further, in subsection III-D, we will present an enhanced implementation of the μ SDN that can support the dynamic modification of RPL parameters.

B. Reducing SDN signaling

As indicated in the Introduction section, as it is still a very common scenario in current indoor use-cases, such as smart buildings ones, we assume that sensor/actuator nodes composing the multi-hop network are static. The support of mobile nodes is left for future works.

Our fundamental concept to reduce the overhead of SDN messages is to avoid frequent and periodic NSU transmissions. Instead, an NSU transmission is triggered only when the nodes observe a state change that might induce a modification of the network topology or flow paths.

Currently, in μ SDN, the information carried in NSU messages includes the node's rank, the number of neighbors, and the *Received Signal Strength Indicator (RSSI)* for each link. The node's rank is derived from RPL and is stable when the network is stationary. Similarly, when sensor nodes are fixed, the number of neighbors changes minimally over time.

In contrast, the RSSI value can exhibit significant fluctuations [12]. However, in most cases, frequent updates are deemed unnecessary, as these fluctuations generally do not result in a change in the routing path. To minimize the number of unnecessary updates due to RSSI fluctuations, we follow a threshold-based approach. The RSSI thresholds are established accordingly with expected ranges of Packet Delivery Ratio (PDR). We defined three PDR ranges: [0, 80%[, [80%, 90%[and [90%, 100%] respectively corresponding to *poor*, *acceptable* and *good* node's communication states. The associated RSSI thresholds could be derived empirically or through observations.

In summary, NSU transmissions are triggered only in the event of changes concerning: (a) the node's rank, (b) the number of neighbors, and (c) the RSSI value exceeding a threshold. Those cases represent situations where the update is considered important, as it may lead to a path change. In

the absence of periodic NSU messages, the controller assumes that the old state of the node is still valid.

C. Minimizing network access delay for periodic traffics

Whenever a node has data to send, it checks if there is a corresponding entry in its flow table. If so, the packet is immediately transmitted. Otherwise, it sends to the controller a *Flow Table Query (FTQ)* packet to request a path. The controller can then respond with a *Flow Table Set (FTS)* configuration message. The path provided in this manner is valid for a fixed duration, defined by the *Flow Table Lifetime (FTL)* parameter. This request is not periodic but rather intermittent, occurring when the node has data to send. The current implementation allows, in case of unavailability of the path at the time the node wants to send data, to delay the transmission until the reception of the path.

We propose to mitigate this delay for scheduled (e.g. periodic) data traffic, which is often the type of traffic draining most IoT resources (bandwidth and battery). We follow a proactive approach through the early sending of FTL requests. Precisely, an FTL request will be generated when 75% of the elapsed duration to the next scheduled data transmission is reached. Consequently, when a node intends to transmit data, the path will already be available and the network access delay will be reduced. In addition, we leverage the knowledge of the data transmission schedule to accordingly adjust the lifetime of the flow table, so that we prevent the expiration of the path's validity before data transmission.

D. Dynamic configuration of RPL parameters

To move towards a dynamic configuration of RPL parameters, we propose to introduce these considerations with μ SDN. As our study is based on intelligent buildings, we have adopted two scenarios with two different operating modes: (a) *Day mode* and (b) *Night mode*. We assume that during the day, when people are present, more data will be collected, in function of human activities. Conversely, in night mode, there is less data and less interference due to the absence of human activity. In this case, an energy-saving mode is envisaged.

The transition from day to night mode can be activated by the SDN controller through the transmission of a **CONF** message to the network nodes. Those messages were defined in μ SDN to initialize various parameters, including the FTL, NSU sending period, I_{min} , I_{max} , and the lifetime of RPL routes. However, dynamic settings using CONF messages are not supported in recent μ SDN implementation.

In enhanced μ SDN, we upgraded the recent μ SDN code with functionalities allowing the dynamic setting of RPL parameters through CONF messages. We focused particularly on the I_{max} parameter and the RPL routes' lifetime, as they have a significant impact on energy consumption in the RPL protocol [2], [3]. Additionally, we enabled the scheduling of CONF messages to support the autonomous switch between day and night modes. This allows for the transition from day to night mode configuration in a completely autonomous and dynamic manner.

IV. PERFORMANCE EVALUATION

A. Simulation methodology

In this section, we detail the simulations conducted under the Cooja simulator of Contiki OS 3.0 to assess NSU reduction and the anticipating establishment of paths for periodic traffic proposals, comparing them to basic RPL and μ SDN. Additionally, we explore the dynamic configuration of RPL through e- μ SDN, considering a Day/Night mode. Table I summarizes all setup parameters.

TABLE I: Simulation parameters

Parameter	Value
Transmission Range	50 m
Interference Range	50 m
Max Bit Rate	9 bit/sec
Radio Medium	UDGM (distance loss)
RPL Mode	Non-storing mode
RPL Route Lifetime	10 minutes
NSU Period	180 seconds
Flowtable Lifetime	10 minutes
DIO_Interval_Doublings	8
Path_Control_Size	0
Delay_DAO	1
DIS_Interval_Timer	60 seconds
DIS_Transmit_Count	5
Objective function	MRHOF

The simulated network, illustrated in Figure 1, is inspired from a real IoT testbed topology deployed for a smart building experiment [13]. We first simulated a smaller infrastructure, called **Topology 1**, which is composed of 7 nodes (represented by a single circle), including 6 sensor nodes and one root node (identified by ID=1). Then, the second studied infrastructure, named **Topology 2**, extends topology 1 with 8 additional nodes (represented by double circles).

We conducted experiments with various periodic data traffic models. The following data generation periods at each node were evaluated: 1, 5, 10, and 30 minutes.

Using these settings, we empirically derived the RSSI thresholds associated to our NSU transmission strategy. Through extensive simulation scenarios, we observed that:

- An RSSI in the range [-25, -10] dBm is associated to a PDR above 90%.
- An RSSI in the range [-55, -25] dBm results into a PDR between 80% and 90%.
- Finally, an RSSI lower than -55 dBm leads to poor communication quality with a PDR below 80%.

We conducted a comparative study between the enhanced e- μ SDN, legacy μ SDN, and basic RPL, considering the following performance metrics.

B. Assessment metrics

SDN Signaling overhead expresses the additional signaling cost induced by SDN control messages. It is evaluated with respect to RPL as a benchmark. Formally:

$$\hat{N} = \frac{N_{SDN}}{N_{SDN} + N_{RPL}} \quad (1)$$

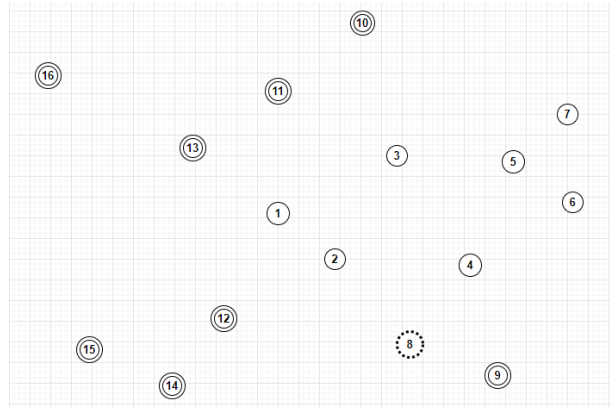


Fig. 1: Network Topologies 1 and 2

where N_{RPL} denotes the total number of RPL control messages, such as DIO, DAO, DIS, and DAO-ACK, generated during a simulation and N_{SDN} represents the amount of SDN messages, including NSU, CONF, FTS, and FTQ.

SDN Energy overhead expresses the additional energetic cost induced by the use of SDN. Similarly to the previous metric, SDN energy overhead, denoted \hat{E} is also expressed proportionally to RPL:

$$\hat{E} = \frac{E_{SDN} - E_{RPL}}{E_{RPL}} \quad (2)$$

where E_{RPL} represents the average amount of energy consumed by all nodes when routing is solely supported by RPL. Energy consumption due to transmission and reception are both considered. E_{SDN} denotes the average consumed energy when the network uses either μ SDN or e- μ SDN.

Packet Delivery Ratio : It is defined as the ratio between the total number of data packets transmitted from source nodes and those received by the root.

As indicated in section III-C, a **Network Access Delay** might be experienced for some data transmissions when the path from a sensor to the root is no longer available.

C. Overhead analysis

The control messages overhead is represented in Figure 2. It illustrates the positive impact of introducing NSU reduction on reducing network overhead. For instance, in the topology 1 traffic scenario with a periodicity of 1 minute, while μ SDN contributes to 28 % of the total network overhead, e- μ SDN has been able to reduce this to 14%, corresponding to an amount of 50 % reduction.

SDN Energy overhead is represented in Figure 3. It demonstrated how the reduction in μ SDN control messages overhead directly influences the energy consumption of the network for both simulated topologies. In a scenario with a 10-minute periodicity for example, in topology 1, μ SDN consumes over 90% more energy than RPL, while e- μ SDN only exhibits a 40% increase compared to RPL. This reduction in energy consumption overhead with e- μ SDN surpasses 50%.

To test the viability of our proposal in a scenario with variations in link quality, and to assess the impact of NSU reduction

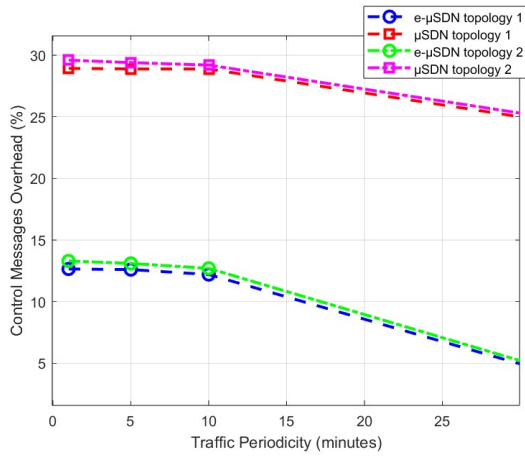


Fig. 2: Control Messages

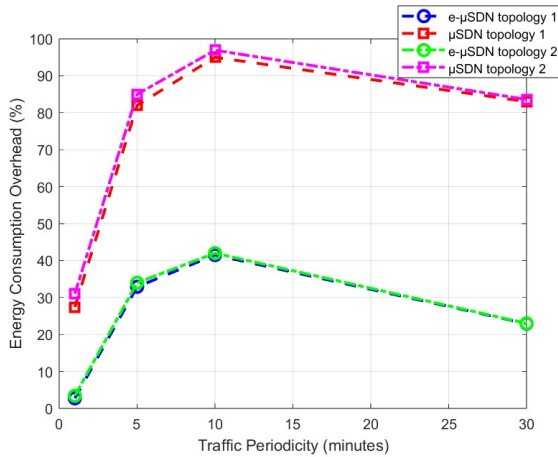


Fig. 3: Energy consumption

in such situations, we re-simulated the same topologies by adding a new node (represented with a dotted circle in Figure 1) capable of generating interference for 15 minutes every 30 minutes. The duration and periodicity of data traffic remained the same. We assessed the overhead of energy consumption for both μ SDN and e - μ SDN. The results confirm the impact of enhanced μ SDN on reducing overhead in terms of energy consumption, even in a scenario with interference.

D. Packets Delivery Ratio

For topology 1, in a scenario without interference, both e - μ SDN and RPL yield the same packet delivery ratio (PDR=1), unlike μ SDN, which provides a slightly different one (PDR=0.99) for a data traffic periodicity of 1 minute. This is due to collisions resulting from the concentration of μ SDN overhead and the intensity of the traffic in question. Under the same scenario, for topology 2 with 15 nodes a packet loss which revolves around 1% (RPL and e - μ SDN) to 2% (μ SDN) was observed for traffic with periodicities of 1, 5, and 10 minutes. The reason for these losses is attributed to interference due to the density of the topology.

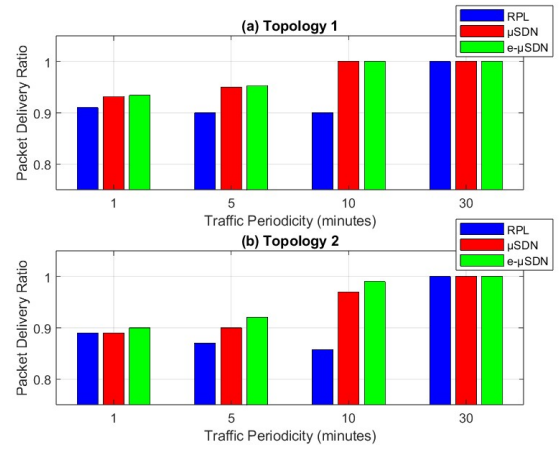


Fig. 4: Packet Delivery Ratio

Figure 4 represents the packet delivery ratio of the scenario with interference. The impact of both μ SDN and e - μ SDN on improving PDR is evident, unlike RPL, whose performance decreases in such a scenario.

The main reason for this is linked to the source routing of μ SDN, which, thanks to its global and up-to-date topology view at the controller, enables the transmission of reliable and current paths to the nodes. In contrast, RPL relies on next-hop routing, guiding packets from parent node to parent node without a global understanding of the topology and with an extended duration for detecting paths with poor quality.

E. Network access delay

To evaluate our proposal, which focuses on anticipating the construction of paths for periodic traffic, we assessed the same simulations for latency and packet delivery ratio. Under μ SDN, for topology 1, the average network access delay is 264 ms for traffic with periodicity of 5, 10, and 30 minutes, and 277 ms for a 1-minute periodicity. For the same periodicity in topology 2, this delay increases to 331 ms. This can be explained by the significant number of requests received by the controller and the increased number of hops, resulting in additional delay. In contrast, e - μ SDN, by anticipating route construction, enables immediate data transmission, resulting in zero delay under all evaluated scenarios.

F. Dynamic configuration of RPL

We now investigate the ability of e - μ SDN to support the dynamic configuration of RPL parameters, by analyzing its performances when switching between day and night modes. To reflect the observation that expected activities in smart buildings by the night are expected to be less intensive than during the day, we studied two traffic models. In **Traffic 1** each node generates data every minute during the day mode, while the data generation period is extended to 10 minutes in the night mode. In **Traffic 2**, the data periodicity is set to 5 minutes and then extended to 30 minutes in the night.

The transition between day and night modes occurs through the transmission of a CONF message, carrying the new values

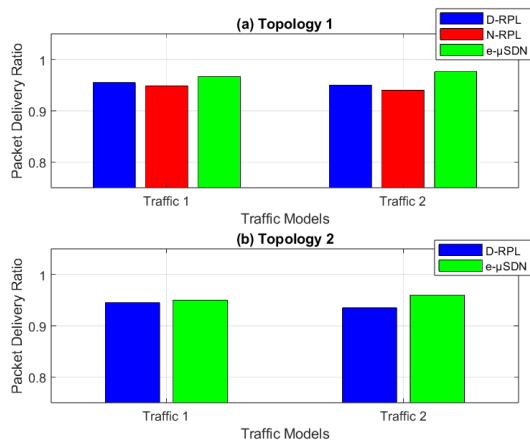


Fig. 5: Packet Delivery Ratio

of I_{max} , RPL routes lifetime, and FTL parameters. Night parameters associated to Traffic 1 and 2 models are specified in Table II, while the day ones are indicated in Table I. RPL with a day configuration noted **D-RPL** and RPL with a night configuration (**N-RPL**) have also been simulated for topology 1, while only D-RPL has been simulated for topology 2.

TABLE II: Nighttime Parameters

Parameter	Value for Traffic 1	Value for Traffic 2
I_{max}	12	12
RPL Route Lifetime	2 hours	6 hours
Flowtable Lifetime	2 hours	6 hours

Two scenarios were simulated. The first scenario involved no interference, while the second scenario introduced interference during the day, following a similar approach as described in the previous evaluation and genuinely associated with human activity. In each case, we evaluated both energy consumption and PDR.

In a scenario without interference, N-RPL consumes less energy than e-μSDN and D-RPL while offering the same Packet Delivery Ratio (PDR). However, in a scenario with interferences, as depicted in Table III, e-μSDN exhibits lower energy consumption compared to D-RPL and N-RPL. This is attributed to the night mode, enabling energy savings and its effectiveness in scenarios with interference during the day. Additionally, e-μSDN provides a better PDR, as shown in Figure 5, illustrating the PDR of D-RPL, N-RPL, and e-μSDN. This is due to its source-based routing, as explained earlier.

TABLE III: Energy Consumption

		Traffic 1	Traffic 2
Topology 1	D-RPL	177.7 J	173.40 J
	N-RPL	172 J	163.09 J
	e-μSDN	160.7 J	163.81 J
Topology 2	D-RPL	1074.6 J	1058.7 J
	e-μSDN	1054.55 J	1049.5 J

V. CONCLUSION AND PERSPECTIVES

This work investigates the case where SDN is used to support adaptive configuration of routing protocol accordingly with the cyclic human activities in smart buildings and we address the challenge raised by the transmission and reception of SDN control messages, which can lead to a significant overhead in term of energy consumption for low-cost IoT devices. To that purpose, we propose three mechanisms that we integrated into μSDN. The first mechanism adjusts the transmission of SDN control messages accordingly with the state of network topology. The second mechanism anticipates path construction for scheduled traffic so that the network access delay is minimized. The last mechanism allows an adaptive configuration of RPL routing protocol to accommodate with different data traffic intensities induced by cyclic human activities in smart buildings. The evaluation of our proposal, so-called e-μSDN, demonstrated its effectiveness.

In future prospects, we will investigate the use of AI techniques to autonomously and dynamically adjust RPL and SDN parameters. We will also construct extensive experiments using a real testbed IoT platform deployed in our lab.

REFERENCES

- [1] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "Rfc 6550: Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.
- [2] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the rpl routing protocol," in *2011 IEEE International Conference on Mechatronics*, pp. 767–772, IEEE, 2011.
- [3] A. Gasouma, K. M. Yusof, A. Mubarakali, and O. E. Tayfour, "Software defined network for energy efficiency in iot and rpl networks," *Soft Computing*, pp. 1–10, 2023.
- [4] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Computer Networks*, vol. 75, pp. 453–471, 2014.
- [5] M. Baddeley, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, "Evolving sdn for low-power iot networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 71–79, IEEE, 2018.
- [6] L. F. da Silva Santos, F. F. de Mendonça Júnior, and K. L. Dias, "μsdn: an sdn-based routing architecture for wireless sensor networks," in *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, pp. 63–70, IEEE, 2017.
- [7] P. Lewis, T. H. Clausen, J. Hui, O. Gnawali, and J. Ko, "Rfc6206: The trickle algorithm," 03 2011.
- [8] T. Theodorou and L. Mamas, "Coral-sdn: A software-defined networking solution for the internet of things," in *2017 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*, pp. 1–2, IEEE, 2017.
- [9] P. Sanmartin, K. Avila, S. Valle, J. Gomez, and D. Jabba, "Sbr: a novel architecture of software defined network using the rpl protocol for internet of things," *IEEE Access*, vol. 9, pp. 119977–119986, 2021.
- [10] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks," in *2015 IEEE conference on computer communications (INFOCOM)*, pp. 513–521, IEEE, 2015.
- [11] M. Kulkarni, M. Baddeley, and I. Haque, "Embedded vs. external controllers in software-defined iot networks," in *2021 IEEE 7th international conference on network softwarization (NetSoft)*, pp. 298–302, IEEE, 2021.
- [12] A. Bildea, *Link Quality in Wireless Sensor Networks*. Theses, Université de Grenoble, Nov. 2013.
- [13] U. Erol, F. Raimondo, J. Pope, S. Gunner, V. Kumar, I. Mavromatis, P. Carnelli, T. Spyridopoulos, A. Khan, and G. Oikonomou, "Multi-sensor, multi-device smart building indoor environmental dataset," *Data in Brief*, vol. 49, p. 109392, 2023.