DTU Library

Synchronized ESP-NOW for Improved Energy Efficiency

Magzym, Yermakhan; Eduard, Aida; Urazayev, Dnislam; Fafoutis, Xenofon; Zorbas, Dimitrios

Published in:
Proceedings of the 11th IEEE International Black Sea Conference on Communications and Networking

Publication date:
2024

Document Version
Peer reviewed version

Synchronized ESP-NOW for Improved Energy Efficiency

Yermakhan Magzym*, Aida Eduard*, Dnislam Urazayev*, Xenofon Fafoutis†, Dimitrios Zorbas*
*Nazarbayev University, School of Engineering & Digital Sciences, Astana, Kazakhstan
†Technical University of Denmark, Denmark
Authors' email: {firstname.lastname}@nu.edu.kz, xefa@dtu.dk

***Abstract*—ESP-NOW is a peer-to-peer wireless communication protocol developed by Espressif Systems for low-power and low-cost in-situ Internet of Things devices. ESP-NOW supports bidirectional communication between multiple transmitters (masters) and multiple end-devices (slaves). It also supports both unicast and broadcast transmissions. This paper proposes a synchronized method, called Sync-ESP-NOW, to improve the energy efficiency of the end-devices by scheduling broadcast transmissions in predefined time intervals. The proposed method creates a synchronized application layer which allows the end-devices to be in *on* mode only when a master's transmission is performed, while they remain in sleep mode to conserve energy for the rest of the time. However, due to the periodic wake-up of the end-devices, a compromise in terms of delay exists. We evaluate the proposed approach through a series of experiments and demonstrate significant improvements in terms of energy efficiency at the end-devices without compromising the packet reception ratio. The proposed approach achieves an up to 96% lower energy consumption considering various packet rate scenarios.**

I. Introduction

The use of measurement, control, and monitoring systems is rapidly increasing in order to gain a deeper understanding and improve the efficiency of physical processes. The advent of Internet of Things (IoT) systems has greatly contributed to this trend, as they offer autonomous monitoring and reporting capabilities. IoT technology has numerous practical applications, including healthcare, construction monitoring, smart homes, and smart agriculture [1].

To address the challenges in deploying and maintaining a vast number of devices in an IoT network, which impact both accessibility and system cost, it is necessary to develop protocols with a more open and generic architecture than traditional wireless networks. A candidate solution for low- and mid-range IoT networks is ESP-NOW, a low-cost and low-power protocol for devices operating at the 2.4GHz ISM spectrum. ESP-NOW is widely used in industrial, smart agriculture, and smart home applications [2]. Additionally, ESP-NOW can be utilized as a supporting and standalone module for network debugging, configuration, firmware upgrades, and real-time control of IoT devices, and enables multiple devices to communicate through low-cost IEEE802.11 transceivers [3].

ESP-NOW has emerged as a cheap and easy to use IoT solution for applications that require connection-less type of communication in order to save energy. Its effectiveness in terms of range and energy consumption has been evaluated in a number of recent publications [4, 5, 6]. It has been found that it can provide hundreds of meters of range in an open space without obstacles and a few dozens meters of range in dense buildings. Apart from that, the long range proprietary mode can slightly outperform conventional IEEE802.11-based solutions in terms of range and packet reception rate at the expense of a higher power expenditure during transmission [6]. However, the authors report an overall much less energy cost compared to connection-full solutions such as the WiFi.

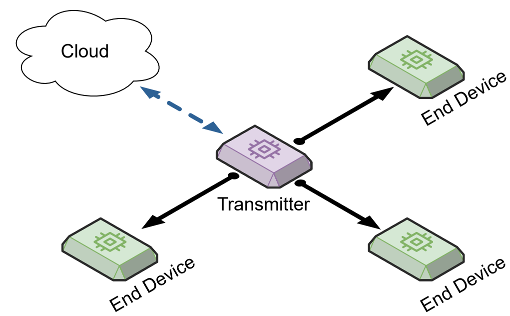

Fig. 1. Broadcast communication scenario.

In this paper, we consider a broadcast communication scenario as it is depicted in Fig. 1. The scenario consists of a master device which periodically forwards data that receives from an external network to a group of slave end-devices (EDs). The transmission is done in broadcast mode so that all EDs receive the data at the same time and minimize delays. Typical applications of such a scenario are the fire-alarm notification systems [7], the localization systems [8], and the clock synchronization systems [9].

In order to tackle the high energy consumption issue that occurs by constantly listening for incoming data, we propose a synchronized version of ESP-NOW which allows EDs to be in receive mode only for short and predefined periods of time while they remain in deep sleep mode for the rest of the time. This is a typical approach followed by known protocols and approaches in the literature in order to have the receivers ready when there is such a need [10, 11, 12]. In the proposed approach, the EDs wake up periodically to synchronize with the master device and receive data. The proposed method creates an application-layer overlay which schedules ESP-NOW packets into a new periodic frame format.

Since transmissions are done at predefined times, the protocol trades delay for energy efficiency. As a consequence, the sleep time must be adapted to the delay requirements of the application. Low delay tolerant applications must allow the transmitter to forward packets more often to the EDs, thus, the sleep period must be shorter. Nevertheless, the energy savings are huge compared to the constant radio on solution.

Summarizing, the contributions of the paper are as follows:

- A synchronized ESP-NOW approach is proposed to significantly reduce the energy consumption of the EDs.
- An application layer implementation of the approach on real hardware is also described, demonstrated, and freely provided to the research community.

II. The ESP-NOW protocol stack

This section briefly presents the main functionalities of the ESP-NOW protocol. The protocol merges functionalities of several OSI layers into a single layer, but at the lower layers, it heavily relies on the native IEEE802.11-1999 standard [13]. However, unlike WiFi which also relies on IEEE802.11, there is no need for a transmitter and a receiver to establish a connection prior to communication, as everything operates on a peer-to-peer basis and in an adhoc manner.

At the physical layer, ESP-NOW is based on the standard IEEE802.11 direct sequence spread spectrum (DSSS) modulation, that is also met in the initial version of IEEE802.11. The default bitrate is equal to 1 Mbits per second, while additionally there is a long-range (LR) version of the protocol which supposedly achieves a longer range at 250 or 500 Kbits per second. The LR version trades range with data rate by using a proprietary modulation. All devices of the network must be set up with the same setting in order to communicate over the LR mode.

At the link and MAC layer, ESP-NOW can be set to one-to-many or many-to-many modes of communication. The default IEEE802.11 CSMA mechanism is used to avoid collisions. Furthermore, it supports two modes of transmission: the broadcast and the unicast mode. In the unicast mode, the transmitter sends a specific packet to a designated ED by including the ED's MAC address in the relevant frame field. In the broadcast mode, the transmitter can send a packet to multiple EDs at once by including a full-bit MAC address in the relevant frame field. No acknowledgment is required for broadcast transmissions.

Two devices can send and receive payloads of up to 250 bytes long via IEEE vendor-specific action frames. Vendor-specific action frames are 802.11 management-type MAC frames, that are used to achieve supervisory functions (e.g. when leaving and joining access points or wireless networks) according to the IEEE802.11 standard [13]. The ESP-NOW packet format is depicted in Fig. 2, according to the currently released implementation[1]. The protocol exhibits an overhead of a total of 43 Bytes.

There is an option to encrypt the payload at the application layer via the widely used CCMP protocol. A device stores a

[1]https://github.com/espressif/esp-now

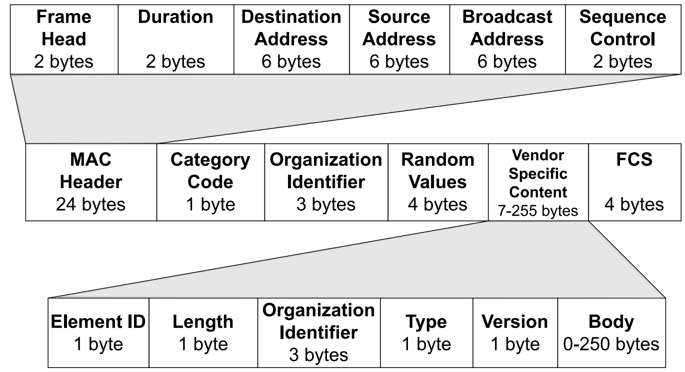

Fig. 2. The ESP-NOW packet format following the IEEE802.11 Vendor-specific action format.

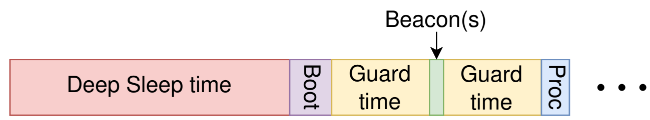

Fig. 3. The Sync-ESP-NOW frame structure (round).

Primary Master Key (PMK) and Local Master Keys (LMK), both 16 bytes long. LMK are encrypted using PMK using the AES-128 algorithm, after that the vendor-specific action frame is encrypted using LMK of the paired device. Broadcast transmissions however are not encrypted by default. The network administrator can provide encryption at the application layer using pre-shared keys.

III. Synchronized ESP-NOW

In this section, we propose Sync-ESP-NOW, a synchronized ESP-NOW approach to reduce the energy consumption of EDs. A scenario with one master and multiple (theoretically infinite) energy-constrained slave EDs is considered, where the master device forwards data to the slave EDs in broadcast mode. The proposed protocol along with the system limitations are discussed in detail.

A. Sync-ESP-NOW Overlay Format

Sync-ESP-NOW employs a repeated overlay application layer frame structure which allows IEEE802.11 data frames to be received in a synchronized way. As depicted in Fig. 3, this frame structure organizes the time in repeated rounds, where consecutive broadcasts of the same packet are performed in each round. The EDs are in active mode only for the duration of the data transmissions plus some extra time to compensate for the boot time and possible clock drifts, while they remain in deep sleep mode for the rest of the time. The transmitter broadcasts the same data multiple times in order to compensate for possible collisions or misses due to the channel path-loss.

Since the data transmissions (beacons) are done periodically, the EDs need to wake up before the transmissions happen and turn their radio on to receive at least one beacon. Due to the synchronization, the clock drift must be taken into account. The clock may drift positively or negatively in between rounds, hence an equal-size guard time is added on the left and on the right of the expected beacons transmission time, respectively. The decision of how long the guard time should be is explained in the next paragraph. Once the beacons are received, a processing time follows, during which the EDs calculate the clock correction and adapt the next wake-up time accordingly. This is done by counting the time since the radio was up until the beacons are received. If that time is less than the guard time, it means that the ED woke up earlier compared to the transmitter's clock. The opposite holds if that time is higher than the guard time. The time difference is added to the sleep time of the next round.

If for some reason all beacons are missed, the ED will go to deep sleep after the end of the second guard time and it will increase the guard time of the next round. If more than two consecutive beacon batches are missed, the ED is considered as desynchronized and it will turn its radio on for the duration of two frames until it receives a new beacon and synchronizes with the transmitter's clock.

The CSMA nature of IEEE802.11 may cause some delays and eventually lead to missed synchronizations if the transmitter finds the medium busy for a long period of time. Because the back-off time of CSMA is in the order of microseconds, these minor delays can be compensated by the guard times which are in the order of milliseconds. Thus, desynchronizations due to transmission delays may happen only in very high congested networks. For that reason, a series of experiments are conducted in Section IV to assess the effect of external interference on the Sync-ESP-NOW transmissions.

B. Energy savings

Keeping the EDs in deep sleep mode leads to high energy savings which are higher the longer the deep sleep period. In order to measure the energy savings due to synchronization, it is important to understand the energy consumption during all steps of the round duration. The following equations show the energy consumption of an ED for the duration of a round (i.e., t), with and without synchronization. The energy consumption is denoted with E_w and E_{wo} for the two cases respectively.

$$E_{wo} = P_{rx} \cdot t, \tag{1}$$

where P_{rx} is the power consumption in receive mode.

$$\begin{aligned} E_w &= E_{boot} + E_{Rx} + E_{slp} \\ &= E_{boot} + P_{rx} \cdot 2 \cdot t_g + (t - t_{boot} - 2 \cdot t_g) P_{slp}, \end{aligned} \tag{2}$$

where E_{boot} is the energy expenditure during boot time, E_{Rx} is the energy expenditure in receive mode, and E_{slp} is the energy expenditure in sleep mode. E_{boot} (as well as the boot time t_{boot}) can be found experimentally while E_{Rx} can be calculated by multiplying the power consumption in receive mode with the time length of two guard times (worst case scenario). Guard times are denoted with t_g. E_{slp} can also be written as the product of the power consumption in sleep mode (i.e., P_{slp}) and the duration of the sleep time.

The guard time depends on the duration of the round or – in other words – on the duration of the sleep time. The relation between the guard time and the round length is expected to be linear because the clock drifts linearly in time (assuming stable environmental conditions). A series of experiments were conducted on a few devices to get an estimation of the absolute drift time for different round lengths and the results are depicted in Fig. 4. The results reveal an almost linear increase of the drift time with higher round lengths. A fit function was also computed for these experiments using the linear regression method. Moreover, the maximum drift time for this specific set of experiments was measured to be up to 35% higher than the fit value. This experimental process can be used as a tool to estimate the drift time at a calibration stage during deployment where the operator can measure the drift of each device independently. For this particular set of devices, the clock drift can be estimated as a function of the round length as follows:

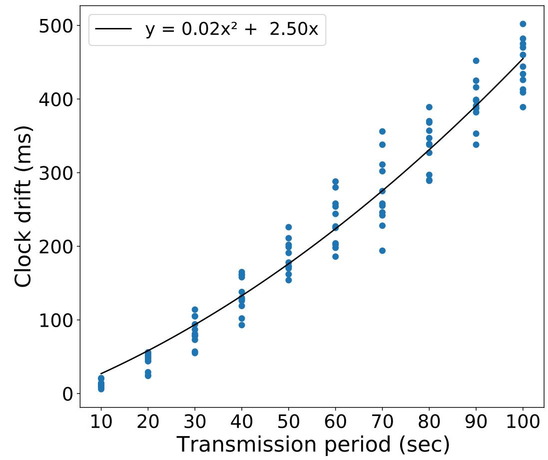

Fig. 4. Absolute clock drift time measured on a set of devices for different round lengths.

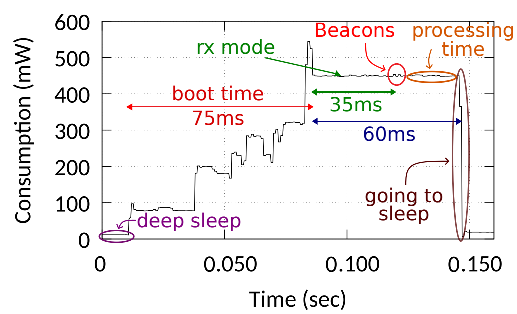

Fig. 5. Receiver's power consumption at different stages.

$$t_g = 1.35 \cdot (0.02t^2 + 2.50t). \tag{3}$$

Given Eq. (1) and (2), the energy savings due to the synchronization are calculated as follows:

$$E_{savings} = \frac{E_{wo} - E_w}{E_{wo}} \cdot 100\%. \tag{4}$$

Fig. 5 illustrates the ED power consumption from the time the device wakes up from the sleep mode until it goes back to it for a round length of 10 seconds. The whole process takes slightly more than 135ms. 75ms are required for the

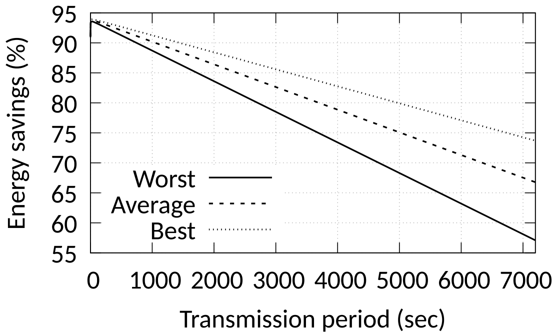

Fig. 6. Energy savings of Sync-ESP-NOW for different beacon transmission periods and experimental drift times (worst, average, and best).

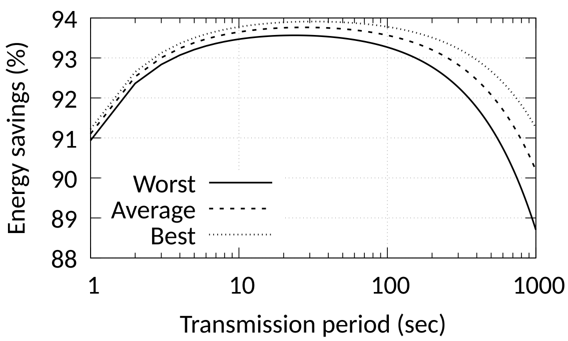

Fig. 7. Energy savings of Sync-ESP-NOW for beacon transmission periods between 1 and 1000 seconds as well as different experimental drift times (worst, average, and best).

ED to boot up and turn the radio on, while another 60ms are needed to receive the beacon, process it, and return to deep sleep. The guard time in this experiment was set to 45ms. The waiting time until a beacon was received was 35ms which indicates a negative clock drift time of about 10ms. Measuring the corresponding times as well as the power and energy consumption values of the previous equations, the synchronized version in this set of experiments was measured almost 94% more efficient than the constant listening version.

Fig. 6 presents the energy savings of Sync-ESP-NOW for different round lengths considering the aforementioned empirical values of the previously conducted experiment. The percentage decreases over time because of the higher drift time that needs to be taken into account but remains high even for long round lengths (e.g., 58% for a 2-hour round length). As Fig. 7 depicts, the energy savings are also high for very short beacon transmission intervals. The savings are maximized when the round length is about 25 seconds.

IV. Evaluation & Discussion of the Results

In order to evaluate Sync-ESP-NOW, a series of experiments were conducted under various conditions. This section presents how the experiments were set up and conducted along with the corresponding results of two sets of experiments.

TABLE I
Experiment Parameters

Parameter	Value
Devices	ESP32-WROOM-32E
Experiment time	∼85min per instance
Transmissions	500
Transmitters	1
Receivers	8
Deployment size	30x30 m
Device positions	see Fig. 8
Tx power	20 dBm
Payload	8 Bytes
Beacons per round	3
Packet rate	1pkt every ∼10 seconds
ESP-NOW mode	Long-Range (250Kbps)
WiFi AP mode	IEEE802.11n
Interference traffic	40Mbps
Firmware version	ESP32 2.0.4
CPU clock speed	240 MHz

Fig. 8. Locations of the devices during the experiments. [M: transmitter position (master), E1-8: ED positions (slaves)]

A. Experiments Setup and Procedure

The experiments are divided into three parts. In the first part, we assess the behavior of Sync-ESP-NOW in terms of Packet Receive Ratio (PRR) at different indoor positions, and thus, at positions with different path-losses. 8 positions were evaluated labeled from E1 to E8 as it is shown in Fig. 8. The positions were chosen based on the Received Signal Strength (RSS): 3 short distance positions (E1, E2, E5), 3 medium distance positions (E3, E4, E6), and 2 further distance positions (E7, E8) without Line-of-Sight (LoS). The first part was conducted in an environment without external interference.

The goal of the second part of the experiments is to examine the effect of external interference on Sync-ESP-NOW perfor-

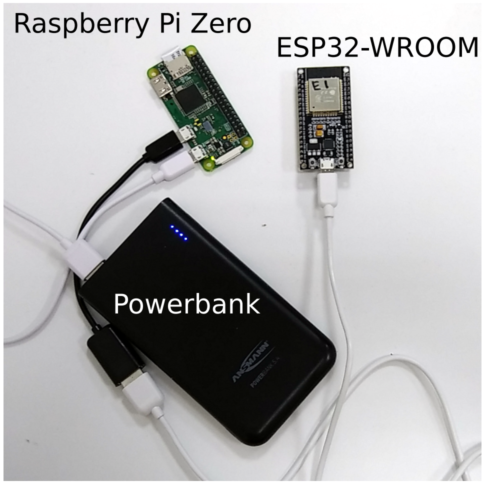

Fig. 9. The equipment used in the experiments for each ED.

TABLE II
EXPERIMENTALLY FOUND TIMES AND POWER CONSUMPTIONS.

Parameter	Value
Wake-up consumption (average)	197 mW
Wake-up time	75 msec
RX mode consumption	450 mW
Deep-sleep consumption (average)	25 mW
Power-off time	<1 msec

mance. 3 devices were used for this purpose – 1 transmitter and 2 receivers. All devices were placed in close proximity to a WiFi access point, while 2 receivers were a few meters away from the transmitter and each other. All 3 devices were in the range of the WiFi network transmitting and receiving packets on the same IEEE802.11 channel. 40Mbps of traffic was generated via video streaming. Sync-ESP-NOW is compared to an approach where the EDs have their radio constantly on (appears as “Always-on” in the results). The only difference with Sync-ESP-NOW is the absence of the sleep time.

In the third part, the energy consumption is measured and compared to the Always-on approach.

The equipment used for each ED in the experiments is shown in Fig. 9. Both approaches were implemented on ESP32-Wroom devices using the C++ programming language[2]. Both the master and the slave devices were equipped with an embedded 2.4GHz antenna. In order to record the number of successfully received packets as well as info about the synchronization, each ED was connected to a Raspberry Pi Zero (RPi) via a USB port. The serial output of the ED was redirected to a log file located at the RPi. The latter was powered up by a powerbank. The transmitter was directly plugged into a powerbank without the presence of a RPi. The log files were retrieved and evaluated after the end of each instance of the experiments. In the first two experiments the transmitter was sending 500 packets with a round length of 10 seconds. Every packet was sent 3 successive times. The overall transmission time per round was less than 4ms. Longer round lengths were tested as well with similar results. Table I summarizes the experiment values while Table II presents times and power consumptions for different stages of the experiments.

[2]The implementation will be publicly available soon.

TABLE III
PACKET RECEPTION RATIO AND NUMBER OF HARD DESYNCHRONIZATIONS FOR DIFFERENT INDOOR POSITIONS.

Position	RSS (dBm)	Sync-ESP-NOW		Always-on
		PRR (%)	Desync.	PRR (%)
E1	-72	100	0	98.4
E2	-68	100	0	100
E3	-79.5	87.4	27	95.6
E4	-79.5	93.6	16	98.1
E5	-63.3	99.6	1	100
E6	-80.5	73.2	55	79.8
E7	-90	58.8	76	62.1
E8	-92	30.6	80	29.8

TABLE IV
PACKET RECEPTION RATIO UNDER THE PRESENCE OF EXTERNAL INTERFERENCE.

Position	RSS (dBm)	Sync-ESP-NOW PRR (%)	Always-on PRR (%)
E9	-64.1	99.6	99.8
E10	-66	99.2	99.6

B. Experiment results

Table III presents the results of the first part of the experiments. We can observe that Sync-ESP-NOW exhibits a considerably high packet receive ratio in positions with higher RSS values, and is visibly close to the Always-on approach. Closer to the transmitter positions present an over 95% PRR compared to further away placed EDs. This is reasonable because the extended path-loss may lead to desynchronizations, and thus, to a slightly lower performance. Indeed, the number of hard desynchronizations for the positions with very low RSS is high even though the PRR is very close (or even higher) than Always-on. As a consequence, the energy savings are also lower in this case because the EDs are less time in deep sleep mode.

Table IV demonstrates the results of the second part of the experiments. Two positions were used with approximately -65dBm of RSS. The results reveal that the presence of external interference does not considerably affect the packet reception ratio of Sync-ESP-NOW. The difference between the non-interference case for positions with the same RSS is not higher than 0.8 percentage units. The average number of hard desynchronizations that Sync-ESP-NOW suffered was 1.5.

C. Energy consumption

Table V presents the energy consumption of a slave device in a 10-minute experiment for different transmission periods. The experiments were conducted in a power analyzer with a constant voltage of 3.75V. The results reveal that Sync-ESP-NOW is more than 90% more energy efficient than the Always-on approach in all tested cases. The actual measured value is

TABLE V
ENERGY CONSUMPTION PER 10 MINUTES FOR DIFFERENT TRANSMISSION PERIODS (TP).

TP (sec)	Energy Consumption (J)	Hard desyn-chronization cost (J)	Energy Savings (%)	Theoretical Energy Savings (%)
5	29.43	1.73	89.10	93.21
10	27.3	2.32	89.89	93.47
30	11.4	6.46	95.78	93.56
60	11.2	14.39	95.85	93.45

also very close to the theoretically calculated savings. We must note that for convenience, the experiments were conducted by not putting all the components of the microcontroller in deep sleep mode, thus in reality, the energy savings could be higher.

V. CONCLUSION & FUTURE WORK

This study presented a synchronized version of the ESP-NOW protocol to improve energy efficiency in broadcast communications. This type of communication can be met in many real-life applications especially in time-synchronization applications. The approach was implemented on a real ESP32 platform and was evaluated in an indoor environment. The results showed a very high reliability in low- and mid-received signal strength positions, while the demonstrated energy savings reached 97% compared to an approach where the end-devices were having the radio constantly on. In cases with very low received signal strength, Sync-ESP-NOW did not perform far from its adversary, however, the energy savings were high. Overall, the proposed approach suffered only from the signal propagation losses at distant locations rather than from the presence of external interference.

In the future, we are planning to design and implement a unicast version of Sync-ESP-NOW using a time-slotted approach. Moreover, we intend to adapt the number of transmissions according to the received signal strength of each end-device.

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Nazarbayev University grant No. 11022021FD2916 for the project "DELITMENT: DEterministic Long-range IoT MEsh NeTworks" and from Samruk-Kazyna JSC for the project "Development of a warning system for repair crews about the approach of a train" implemented by Nazarbayev University for National Railways company JSC grant No. 280921SK0201.

REFERENCES

[1] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241–261, 2019.

[2] H. A. Nguyen, L. V. Nguyen, and Q. P. Ha, "IoT-enabled dependable co-located low-cost sensing for construction site monitoring," in *37th International Symposium on Automation and Robotics in Construction*, International Association for Automation and Robotics in Construction (IAARC), 2020.

[3] G. Amponis, T. Lagkas, D. Karampatzakis, P. Radoglou-Grammatikis, V. Argyriou, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Efficient peer-to-peer unicasting for VANET architectures via enhanced monolithic protocols," in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1–8, IEEE, 2022.

[4] K. Khanchuea and R. Siripokarpirom, "A multi-protocol IoT gateway and WiFi/BLE sensor nodes for smart home and building automation: Design and implementation," in *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pp. 1–6, IEEE, 2019.

[5] M. I. Labib, M. ElGazzar, A. Ghalwash, and S. N. AbdulKader, "An efficient networking solution for extending and controlling wireless sensor networks using low-energy technologies," *PeerJ Computer Science*, vol. 7, p. e780, 2021.

[6] D. Urazayev, A. Eduard, M. Ahsan, and D. Zorbas, "Indoor Performance Evaluation of ESP-NOW," in *2023 IEEE International Conference on Smart Information Systems and Technologies*, pp. 1–5, IEEE, 2023.

[7] K. Muheden, E. Erdem, and S. Vançin, "Design and implementation of the mobile fire alarm system using wireless sensor networks," in *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 000243–000246, IEEE, 2016.

[8] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, and M. Atri, "A survey of localization systems in internet of things," *Mobile Networks and Applications*, vol. 24, pp. 761–785, 2019.

[9] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," *Handbook of sensor networks: Algorithms and architectures*, pp. 199–237, 2005.

[10] LoRa Alliance Technical Committee, "LoRaWAN™ 1.0.4 Specification." lora-alliance.org/resource_hub/lorawan-104-specification-package, 2018. Online; accessed 14-Mar-2023.

[11] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things," *Computer Communications*, vol. 153, pp. 1 – 10, 2020.

[12] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, "TDMA-based MAC protocols for vehicular ad hoc networks: a survey, qualitative analysis, and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2461–2492, 2015.

[13] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.