



Performance Analysis of IEEE 802.15.4 Bootstrap Process

Alberto Gallegos Ramonet ^{1,*}  and Taku Noguchi ² 

¹ Graduate School of Technology, Industrial and Social Science, Tokushima University, Tokushima 770-8506, Japan

² College of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan

* Correspondence: alramonet@is.tokushima-u.ac.jp

Abstract: The IEEE 802.15.4 is a popular standard used in wireless sensor networks (WSNs) and the Internet of Things (IoT) applications. In these networks, devices are organized into groups formally known as personal area networks (PAN) which require a bootstrap procedure to become operational. Bootstrap plays a key role in the initialization and maintenance of these networks. For this reason, this work presents our implementation and performance analysis for the ns-3 network simulator. Specifically, this bootstrap implementation includes the support of three types of scanning mechanisms (energy scan, passive scan, and active scan) and the complete classic association mechanism described by the standard. Both of these mechanisms can be used independently by higher layers protocols to support network initialization, network joining, and maintenance tasks. Performance evaluation is conducted in total network association time and packet overhead terms. Our source code is documented and publicly available in the latest ns-3 official release.

Keywords: association; scanning; beacon-enabled mode; bootstrap; IEEE 802.15.4; IoT; LR-WPAN; ns-3; simulations; Zigbee; WSN; network tools



check for updates

Citation: Gallegos Ramonet, A.; Noguchi, T. Performance Analysis of IEEE 802.15.4 Bootstrap Process. *Electronics* **2022**, *11*, 4090. <https://doi.org/10.3390/electronics11244090>

Academic Editors: Nurul I. Sarkar and Juan-Carlos Cano

Received: 4 November 2022

Accepted: 6 December 2022

Published: 8 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The IEEE 802.15.4 is a standard conceived for low-rate wireless personal area networks (LR-WPANs) and used in transceivers that operate using low data rates and with low energy consumption requirements. This LR-WPAN standard defined a physical (PHY) layer and a media access control (MAC) layer and has gone through over 30 amendments to date. IEEE 802.15.4-2011 std. compliant devices (or older) should support two modes: the beacon-enabled mode and the non-beacon-enabled mode. In the non-beacon-enabled mode, data transmission is asynchronous, using a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. In the beacon-enabled mode, data transmissions are synchronized with the help of a coordinator and a slotted version of the CSMA/CA, which allow radios to have sleep periods that save energy. Later revisions of the standard (IEEE 802.15.4e-2012) have notably introduced more advanced MAC mechanisms, such as the deterministic and synchronous multichannel extension (DSME) and time slotted channel hopping (TSCH), considered by many today as state-of-the-art MAC mechanisms. However, most commercial implementations of the standard today still use the MAC mechanisms introduced early in the standard [1–3] either for practical or financial reasons. Examples of this include popular protocols stacks (e.g., Zigbee 3.0 [4] and Thread 3.0 [5] specifications) which only work on top of IEEE 802.15.4 standard (2011 or older). As a consequence, these LR-WPAN legacy MAC modes are still extensively used to develop new applications and proposals.

In this paper, we present a substantial extension to the ns-3 network simulator [6] IEEE 802.15.4-2011 module. Specifically, we extend the capabilities of the module to support three types of network scanning mechanisms: energy detection, passive scanning, and active scanning. Most importantly, we extend this module to support the IEEE association mechanism as described by the standard. Scan and association when used combined enable

the initialization of networks in a process known as bootstrap. The association mechanism is essential to realistically start and configure networks consisting of hundreds of nodes. Likewise, it enables maintenance and administrative functions often used by routing protocols operating on higher layers (e.g., tasks involving leaving or joining adjacent networks). Our review of the literature indicates that this is the first publicly available extension of its kind for the ns-3 network simulator IEEE 802.15.4 module. This proposal has been fully documented, freely available and integrated into the ns-3 simulator official release.

The remainder of this paper is organized as follows: Section 2 provides the background information and the motivations for this study. Section 3 contains descriptions on ns-3 capabilities and implementation details of its IEEE 802.15.4-2011 module. Section 4, describes in detail our scan and association mechanisms implementation. Finally, in Section 5, we present our evaluations followed by our conclusions. A table of terms used in this work can be found in the abbreviations Section at the end of this paper.

2. Background and Motivations

Network initialization is a simple concept in itself; in the IEEE 802.15.4 std., devices must be grouped before performing any communications tasks. In other words, devices must belong to a logical group formally known as a personal area network (PAN) to communicate. In reality, this simple procedure in IEEE 802.15.4-2011 (or earlier) can involve a lengthy scanning of spectrum channels and the exchange of several command frames to complete an association of a single pair of device. In situations where many devices participate in this process simultaneously, performance issues can quickly arise because of the shared communication channel. In network simulations, the association mechanism is often overlooked. This is because members of a PAN can be simply predefined before the beginning of the simulation. This, however, skips a step that in some real LR-WPAN network implementations is unavoidable. For example, routing protocols, such as the one found in a Zigbee stack, uses this MAC association to join, leave, and form subnetworks. Although the concept of bootstrap (network initialization) is not often explored in the literature, it has a measurable cost on the initialization and maintenance of this type of networks and in some cases, it has significant performance implications to higher layer protocols.

In the development of tools and applications for IEEE 802.15.4, the operative system Contiki-ng [7] and its simulator Cooja are considered by many as state of the art for development of IoT solutions. However, this work is an extension specifically intended to increase the capabilities of the ns-3 simulator which has a different focus from the Contiki-ng project. Ns-3's simulated protocol stack uses IEEE 802.15.4-2011 (PHY and MAC: layers 1 and 2) and similarly to Contiki-ng is capable of running 6lowpan, IPV6 and the routing protocol for low-power and lossy networks (RPL: layer 3) on top of its MAC layer. Unlike Contiki-ng, ns-3 can be used to develop other existing popular communication stacks, such as Zigbee pro, Zigbee IP, and Thread, which specifically require an IEEE 802.15.4-2011 standard or older. It is worth mentioning that protocol stacks such as Zigbee pro provides an alternative routing implementation to RPL that is non-IP dependent. Therefore, ns-3 users can use the work described in this paper to support the development of their own non-ip dependable routing protocols similar to Zigbee pro.

On the other hand, Contiki-ng focuses on a more advanced but more complex protocol stack mainly formed by IEEE 802.15.4e-2012 (TSCH), 6lowpan, IPV6 and RPL. Its main focus is the development of its OS to support multiple hardware platforms and to a lesser degree the development of its simulator. Cooja is able to execute complete instances of Contiki-ng, which ensures a closer match between simulations and real world deployments. However, this comes with the cost of slow performance, making simulations over 100 nodes impractical [8]. In this aspect, ns-3 is different in the sense that is constantly optimized and can be configured to work in parallel; reportedly, it has been used in distributed simulations consisting of up to one billion nodes [9]. Similar projects to Contiki-ng, such as OpenWSN [10] and its simulator OpenSim, exist. OpenWsn is not an OS like Contiki-ng, but it has a similar focus to implement a hardware usable protocol stack.

The ns-2 network simulator [11] is probably one of the first simulated implementations of the IEEE 802.15.4 standard. The main problem with this implementation is that it was officially discontinued in approximately 2010. In the past, the ns-2 simulator had significant contributions from many developers; however, it lacks consistency in comparison to ns-3, which mandates a coding style and source code review from all its contributors. Consequently, ns-3 is a well-maintained and well-documented open-source simulator that affords greater clarity than ns-2. Other notable mentions include the OMNET++ simulator which includes support for IEEE 802.15.4 via the Castalia [12] framework. This framework supports the possibility of employing multiple MAC layers that can operate with the 802.15.4 PHY layer. However, as indicated in its manual (v3.3), rather than providing support for indirect transmissions in its 802.15.4 MAC layer (and consequentially association), Castalia supports only direct and GTS transmissions, and no bootstrap mechanism is specified. Additionally, Castalia is distributed as an independent module that runs on top of OMNET++; therefore, the framework is not officially maintained or supported by OMNET++ in any way. While not officially discontinued, Castalia has not been maintained in years and only runs in older versions of OMNET++. Finally, commercial simulators, such as OPNET, QualNet, and NetSim, are alternative options to develop solutions using IEEE 802.15.4. However, simulators such as these require a license, which makes it a restrictive option for those looking to extend the base code or distribute their proposals freely and without access restrictions.

3. The IEEE 802.15.4 Standard

Initially, the IEEE 802.15.4 standard was created as a solution for networks with a low transmission rate and low energy consumption. Devices using this standard are capable of transmissions without the need for complex configurations (non-beacon-enabled mode). However, the real energy-saving features found in the standard are achieved by transmitting data using synchronization in a MAC mode known as the beacon-enabled mode. In both modes, the association is required before devices become operational in a Personal Area Network (PAN). A designated capable device functions as the PAN coordinator which provide services to other devices associated with it. Coordinators must be fully functional devices (FFDs), whereas other participant devices in the network can be either FFDs or reduced-functional devices (RFDs). The network can be extended by connecting multiple coordinators, but only a single device can act as the PAN coordinator. Non-beacon mode devices are asynchronous while in the beacon mode. The PAN coordinator achieves synchronization with the other devices by periodically transmitting beacon frames. The interval between beacon frames (BI) is divided into an active period followed by an optional inactive period often used by transceivers in this standard to deactivate and conserve energy. Equation (1) shows the length of BI.

$$BI = aBaseSuperframeDuration \times 2^{BO}, \quad (1)$$

where the beacon order (BO) is an unsigned integer value between 0 and 14, and *aBaseSuperframeDuration* is a constant defined in the standard as 960 symbols. In the standard, all periods are represented in units of symbols. A symbol can be defined as a waveform with a unique phase and magnitude: in essence, a state during a fixed time. Depending on the modulation used to interpret these waveforms, a symbol can represent the encoding of one or several bits of information. For example, in optional offset quadrature phase-shift keying (O-QPSK) modulation used by one of the PHYs in the IEEE 802.15.4-2011 standard, each symbol is interpreted as four bits of information that can be transmitted in a period of 0.016 ms. The structure of the active period is distributed to all the PAN devices via the beacon frames. This information is known as the superframe structure. The superframe duration (SD) is given by Equation (2).

$$SD = aBaseSuperframeDuration \times 2^{SO}, \quad (2)$$

where the superframe order (SO) is an unsigned integer between 0 and 14, but it must be less than or equal to the BO. Simply put, the SD is the length of the active period formed by 16 time slots, distributed between the time used for the transmission or reception of the beacon, the contention access period (CAP), or the contention-free period (CFP). During the CAP, devices contend to transmit data using a slotted-CSMA/CA algorithm. CFP is an optional period that when present must be preceded by a CAP. CFPs are subdivided into variable-size guaranteed time slots (GTS), which are designated to some devices for a fast-track transmission without the need for contention. Only seven GTSs can be assigned to a single CFP, and its size directly correlates to the number of GTSs assigned and the maximum size of the CFP used. Finally, time slots are subdivided into manageable units known as *backoff periods*, which comprise 20 symbols each. It is also worth noting that beacons are always transmitted during the first time slot of the superframe. The superframe structure described above is summarized in Figure 1.

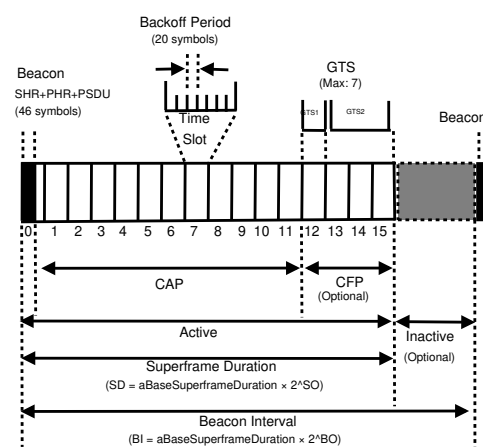


Figure 1. IEEE 802.15.4-2011 time division according to a superframe structure used in beacon-enabled mode.

3.1. Data Transmissions

Data transmissions in the standard can be performed in three distinct ways:

- **Direct Transmissions.** When using direct transmissions and the beacon-enabled mode, data frames can be sent during the CAP portion of the superframe using the slotted CSMA/CA algorithm. Alternatively, if non-beacon enabled mode is in use, data frames are transmitted using a non-slotted CSMA/CA algorithm. If required, data transmissions can also be acknowledged (ACK). ACK frames are sent without using CSMA/CA. Direct transmissions originating from the coordinator are not clearly defined in the standard [3]; however, manufacturers include this possibility in their transceivers (e.g., transfer types for NXP 802.15.4 transceivers [13]).
- **Indirect Transmissions.** In this transmission type, the coordinator enqueues data in its *pending transaction list* until polled by a device. Devices collect this data with the help of beacons. In the beacon-enabled mode, when a device receives a beacon and becomes aware of the awaiting data, it sends a *data request command* to the coordinator. The coordinator acknowledges the data request and searches for its buffered data; if available, it sends a data frame, which the receiving device may acknowledge. Alternatively, devices might poll beacons directly from a coordinator in the non-beacon-enabled mode. Coordinators ignore poll requests if running in the beacon-enabled mode. Transmission of data using indirect transmissions is useful when devices need to sleep for long periods to conserve energy, and when the response time is not a concern. Indirect transmissions are indispensable for buffering command frames in the classic PAN association, as described by the standard. Indirect transmissions for command frames in the association mechanism is covered by this work, but indirect transmissions for data frames are outside the scope of this paper.

- GTS Transmissions. These are transmissions using reserved time slots of the super-frame’s CFP., and they are not supported by the current implementation described in this paper.

3.2. IEEE 802.15.4 Primitives

The IEEE 802.15.4 standard use *primitives* for communication between layers. A compliant IEEE 802.15.4 device uses primitives for both the PHY and MAC layers. The standard supports four basic types of primitives:

- Request. Initiate an event or request a service.
- Indication. Used to indicate when an event occurred.
- Response. Completes a procedure of a previously invoked indication primitive.
- Confirm. This primitive is set off as a result of a previous request of service.

In the PHY layer, primitives are organized into the physical layer data (PD) and the physical layer management entity (PLME) primitives. Likewise, in the MAC layer, primitives are organized into two groups: the MAC common part sublayer (MCPS) and the MAC sublayer management entity (MLME). In our implementation, we focused on the inclusion of MAC primitives that enable bootstrap. The class diagram in Figure 2 shows the classes that form the ns-3 LR-WPAN module and its relationship with other important classes in ns-3. Some of our contributions are highlighted in the figure.

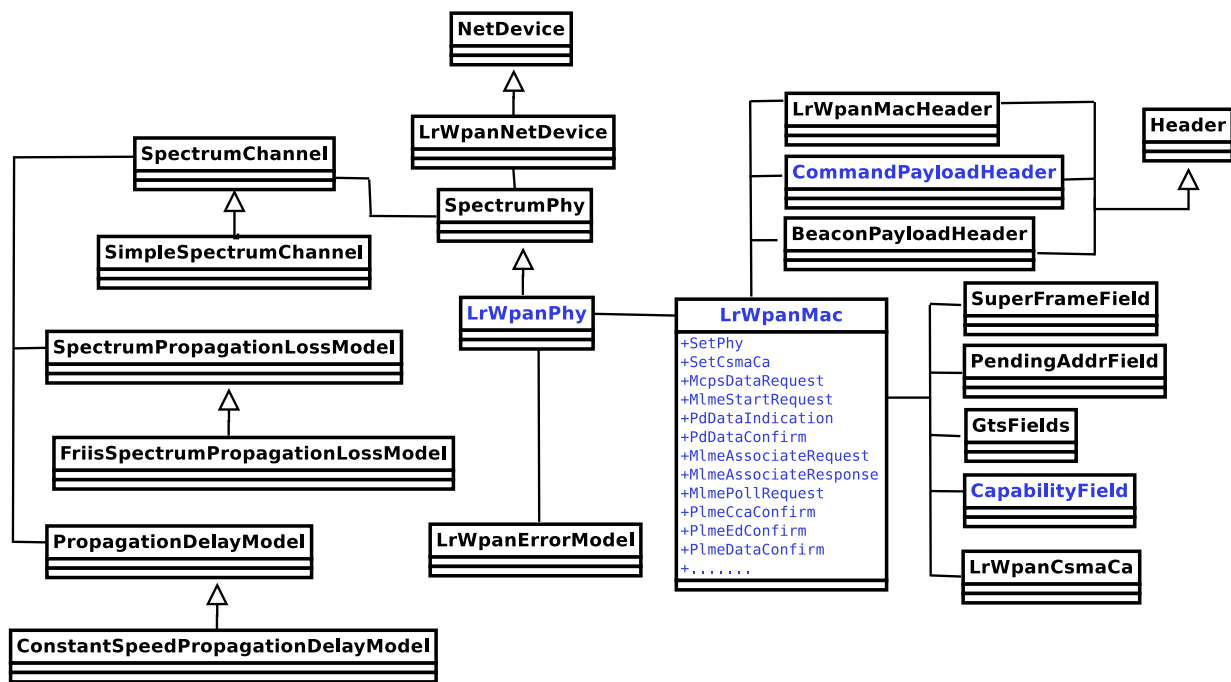


Figure 2. Simplified UML class diagram for ns-3 LR-WPAN module (IEEE 802.15.4) and relationship with other classes. Some of our contributions are highlighted.

The Ns-3 LR-WPAN module is organized into several classes that represent multiple elements of its PHY and MAC layers. The **LrWpanPhy** class was created by Gary Pei based on the initial **SpectrumPhy** class developed by Nicola Baldo et al. [14]. This class provides the basic functionalities of the physical layer in the standard. Implementation details of the **LrWpanPhy** class are outside the scope of this work. Descriptions on similar ns-3 PHYs based on the **SpectrumPhy** abstract class are extensively covered by other authors’ work [15,16]. In this work, **LrWpanPhy** was modified marginally to support pagination which was introduced in the 2011 revision of the standard and is extensively used by the scanning primitives to dynamically switch frequency bands and channels. The **LrWpanMac** class is supported by the **LrWpanPhy** class and contains the MAC layer functionalities described in previous sections. In this work, this class is significantly

extended to include the implementation of all primitives that add support for scanning and association mechanisms. Descriptions of the added primitives are found at the end of this section. The classes **SuperframeField**, **PendingAddrField**, and **GtsFields** provide the superframe substructures used by the MAC layer beacon-enabled mode as described in previous sections. The **CapabilityField** class was added in this work to provide the structure that describes the operational capabilities of a device (security, power, Rx on when idle, etc.). This structure is used in the frame transmitted in association with request commands. The **LrWpanCsmaca** class is the heart of the MAC layer in ns-3, representing both slotted and un-slotted CSMA/CA. Detailed descriptions of the beacon mode and slotted CSMA/CA mechanism details for ns-3 are not covered in this work, but they can be found in our previous work [17].

Finally, classes **LrWpanMacHeader**, **BeaconPayloadHeader**, and **CommandPayloadHeader** provide the MAC layer with the necessary frame headers to support the transmission of data frames, beacon frames, and command frames, respectively. In particular, **CommandPayloadHeader** was implemented in this work to provide the frame headers transmitted during the association process (association request, data request, and association response commands).

The following is a brief description of all MAC primitives used by the scan and association mechanisms implemented in this work:

MLME-ASSOCIATE.Request allows a device to request an association with a coordinator or PAN coordinator. An association is required to transmit data frames between a device and a coordinator. An association request is typically preceded by a scan request to search for the best channel/coordinator for association.

MLME-ASSOCIATE.Confirm is used to inform the next higher layer of the initialization of the association process, regardless of whether the request is successful or unsuccessful.

MLME-ASSOCIATE.Indication is used by a coordinator to indicate the reception of an association request command.

MLME-ASSOCIATE.Response is a primitive that is triggered by a higher layer in response to an **MLME-ASSOCIATE.Indication** primitive.

MLME-SCAN.Request is used by a higher layer to initiate a scan on a given number of channels for a set duration. The results are typically used to locate or request beacons transmitted by coordinators or to measure the current energy state on a given set of channels.

MLME-SCAN.Confirm reports the results of a channel-scan request. The results of passive or active scans are organized in a list of PAN descriptors. These descriptors represent the information extracted from each received unique beacon.

MLME-COMM-STATUS.Indication is a primitive that is used to inform a coordinator next higher layer of the result of the response to an association request. The response to an association request can be either a success or failure. If the indication is a failure, a reason is given. Multiple reasons could exist for failure: the coordinator was not accepting requests, the request expired in the pending transaction list, the channel was busy when the coordinator attempted to respond to the request, acknowledgments were not received at any given point of the response, etc.

MLME-POLL.Confirm is a primitive used during indirect transmissions to inform the next layer of a coordinator device of the results of transmitting a frame obtained (polled) from the pending transaction list where frames are queued (e.g., indirect data frames or association response commands).

Existent primitives in ns-3 are summarized in Figure 3. The implemented primitives in this work are highlighted. Descriptions of other primitives not mentioned here can be found directly in [3].

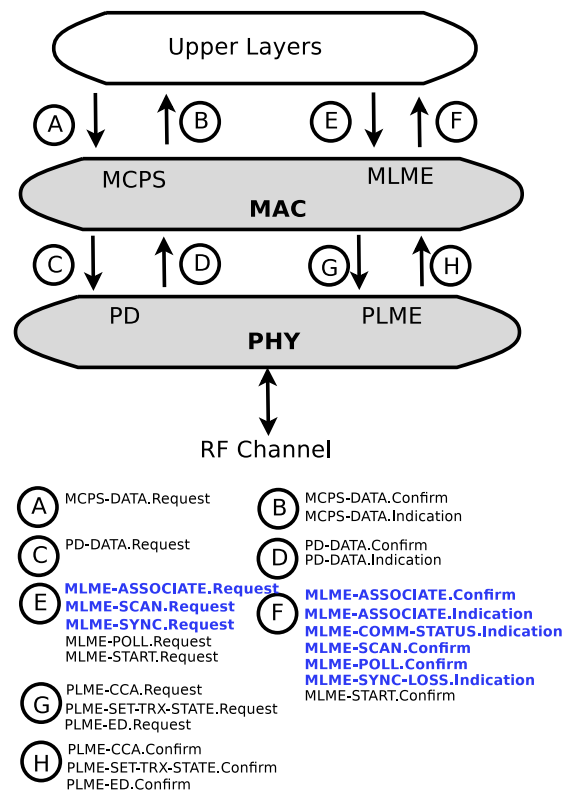


Figure 3. Ns-3 IEEE 802.15.4 layers and its primitives.

4. Bootstrap: Scan and Association

In IEEE 802.15.4, networks are capable of self-initializing (bootstrap) into groups known as PANs. All devices must belong to a PAN before attempting to transmit data frames. The initialization of PANs typically involves two independent processes: channel scanning and coordinator association.

4.1. Channel Scanning

A device scans a channel for one of two reasons: attempting to find coordinators in the channel or measuring the channel peak energy before initiating a network and starting transmitting beacon frames on the link. The standard supports multiple scanning types regardless of the scan type used. The scanning process accepts two main parameters: a channel list and scan duration. In the scanning process, a PHY switches to each channel in the channel list and scans it for the duration specified in the *scanDuration* parameter. In the standard, a $scanDuration = aBaseSuperframeDuration \times (2^n + 1)$, *aBaseSuperframeDuration* is 960 symbols and *n* is a value between 0 and 14. In this proposal, we added support for three types of scans:

- **Active Scan.** In this scan, a device sends *beacon request command* frames to the PAN coordinators on a set number of channels. Coordinators receiving these frames must be configured in the non-beacon mode. Coordinators set in beacon mode ignore these requests. Coordinators who accept the request respond with a beacon frame. The device that initiates the scan gathers information from the collected beacons and organizes this information into a list of *PAN descriptors*. These PAN descriptors are later used to choose the most suitable coordinator during the association process.
- **Passive Scan.** In the passive scan, beacon request command frames cannot be received. Devices assume the existence of coordinators already transmitting beacon frames and automatically scan a set number of channels looking for beacon frames. Similar to the active scan, the information from beacons is gathered and organized in a list of PAN descriptors for later use by other higher layer primitives (e.g., trigger the initiation of the association process).

- Energy Detection (ED) Scan. Energy scans are used to search for traces of energy on a given set of channels within a specific time. The results of an energy scan are stored in a PAN descriptor and provided to the next higher layer.

In addition to the scan types listed above, the standard also describes the use of *orphan scans*. These scans are used to reinitialize the association when a device has lost the track of a coordinator (lost synchronization). Our implementation did not include support for orphan scans, but they are considered for future extensions.

As mentioned previously, scans are often used in combination with the association procedure. Figure 4 shows the sequence involved in bootstrap. In the figure, the coordinator is assumed to be set in the beacon-enabled mode. In the association shown in this figure, devices use passive scans to capture information from incoming beacons and choose the most suitable coordinator with which to associate. The coordinator selection details are not covered by the standard; however, the value of the link quality indicator (LQI) contained in the PAN descriptor is often used to make this choice. LQI is a metric used by IEEE 802.15.4 which measures the error in the incoming modulation of successfully received packets (i.e., the quality of received beacons) [18]. LQIs are contained in the collected PAN descriptors and can only be obtained with passive and active scans. A link with an LQI of 255 is considered to be of high quality, whereas any value below 127 is considered to indicate a poor quality link.

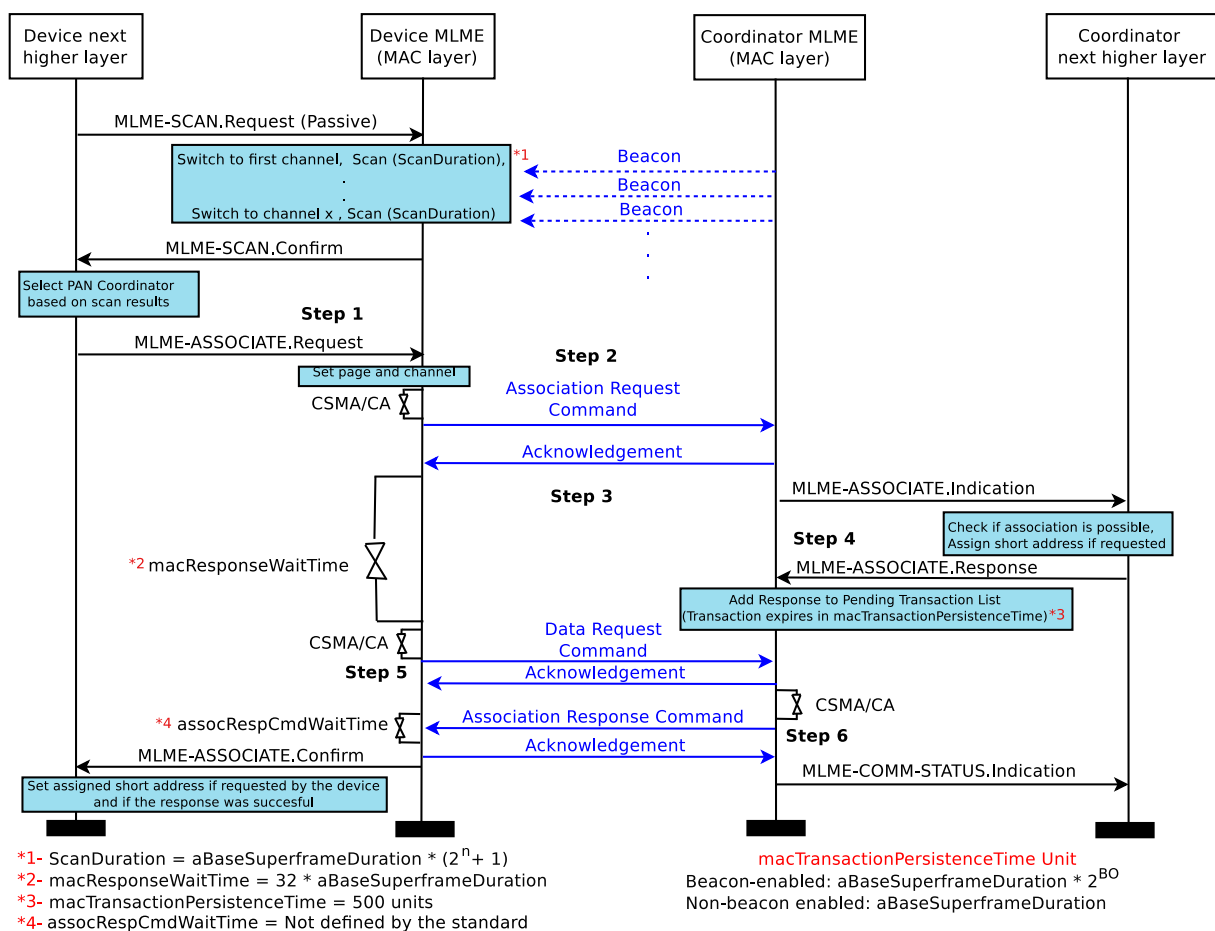


Figure 4. Message sequence chart for bootstrap: passive scan and association.

4.2. Association

Figure 4 shows a typical association request.

Step 1. The association process initiates when an MLME-ASSOCIATION request is issued from the device MAC layer. The chosen page and channel are set according to the parameters in the request. As previously mentioned, the channel and page values

are typically chosen from the results obtained during the scanning process but can be manually set.

Step 2. After establishing the page and channel, the device sends an *association request command* to the chosen coordinator. The device uses the extended MAC address to transmit this request; in fact, all the request and response command frames used in an association process use extended MAC addresses. Upon receiving this request, the coordinator sends an acknowledgment to confirm reception of the association request command.

Step 3. Once the acknowledgment described in the previous step is received, the device is set to enter waiting mode for a duration of *macResponseWaitTime*. The purpose of this pause is to allow the coordinator to process the association request. While the device waits, the coordinator communicates to its next higher layer the reception of the associated request command through an MLME-ASSOCIATION.Indication primitive. Although this is not covered by the standard, in the next higher layer, the coordinator must determine whether the request should be accepted. A coordinator can reject the request if it has reached its association capacity limit or if it is not accepting requests for other reasons (these reasons are decided by the implementer of the next higher layer). Likewise, if specified in the association request, in this step, the coordinator can assign a short address to be used by the device for communication with the coordinator following the association process.

Step 4. After the coordinator has processed the association request, it communicates its decision to its MAC layer through an MLME-ASSOCIATE.Response primitive. The response is then added to the coordinator pending transaction list, where collection by a poll request is awaited. This process is reminiscent of indirect transmissions of *data* frames because it is in fact an indirect transmission. The difference between an indirect transmission used in data frames and that used in *association response command* frames lies in the addressing modes. Indirect transmissions in the association process use the extended address mode exclusively, whereas data frame indirect transmissions can use either short or extended address modes.

Step 5. In this step, *macReponseWaitTime* expires, and the device transmits a data request command frame to extract the associate response from the coordinator's pending transaction list. When the coordinator receives the data request command, it first checks whether it has pending responses destined for the requesting device. If data are pending for the requesting device, it transmits an acknowledgment in response to the data request command.

Step 6. Following the transmission of the acknowledgment in the previous step, the coordinator extracts the response from its pending transaction list and sends it back in an *association response command* frame to the device. Once the device receives this association response command frame, it acknowledges its reception and reports the association result to its higher layer using an MLME-ASSOCIATE.Confirm primitive. If the association response is positive and the device requested a short address assignment in its initial association request, it sets the new short address assigned by the coordinator. Otherwise, it continues using its MAC extended address. When the coordinator receives this last acknowledgment, it informs its next higher layer about the conclusion of the association procedure using an MLME-COMM-STATUS.Indication primitive. An *assocRespCmdWaitTime* is used by a device to react to any missing *association response commands*. If this command is not received within the configured time, the association process is concluded with an MLME-ASSOCIATE.Confirm containing a negative status (NO DATA).

5. Evaluations

In this paper, we described our proposal to extend the ns3 network simulator (IEEE 802.15.4) MAC layer to support scanning and association mechanisms. This proposal prompted us to devise experiments demonstrating the limitations and overall performance of the classic association procedure in IEEE 802.15.4.

5.1. Scanning Experiments

The MAC layer energy scanning mechanism is intrinsically related to the physical layer ED procedure. In this first experiment, we test the scanning capabilities of ns-3 by observing its ED results. Energy scanning is defined as the maximum energy value registered from a series of individual ED within a given duration specified by the user. A single ED consists of an integer value ranging from 0 to 255. The minimum ED value (0) indicates that the average receiver power in a period of 8 symbols (128 μ s in O-QPSK) is less than 10 dB above the Rx sensitivity. The maximum ED value (255) shall at least span a ratio of 40 dB above Rx sensitivity. These thresholds, in addition to the default Rx sensitivity of -106.58 dBm which only accounts for the thermal noise, are used by ns-3 to determine the ED value (Figure 5). Adjusting the Rx sensitivity and the ED thresholds makes it possible to optimize the scanning capabilities of devices. Such adjustments are not supported in the current IEEE 802.15.4 PHY implementation of ns-3. The inclusion of these characteristics is beyond the scope of the current work; however, their inclusion is considered for future works.

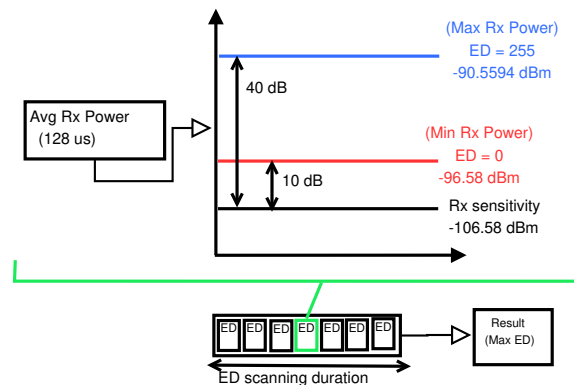


Figure 5. Ns-3 energy scanning and ED procedures.

We used a single node scanning at different distance intervals with increases of 5 m on the x-axis. Two coordinators were configured to generate beacons on the same channel using fixed positions at 0 and 100 m, respectively. Devices use ns-3’s **LogDistancePropagationLossModel** to observe the effects of node separation on the energy scan. Figure 6 shows the results of this experiment. As expected, the maximum energy value (255) was detected at the points where the coordinators were placed. The registered energy value decreases as the scanning node moves away from the coordinators reaching its lowest point in the experiment at 50 m for the default Rx sensitivity.

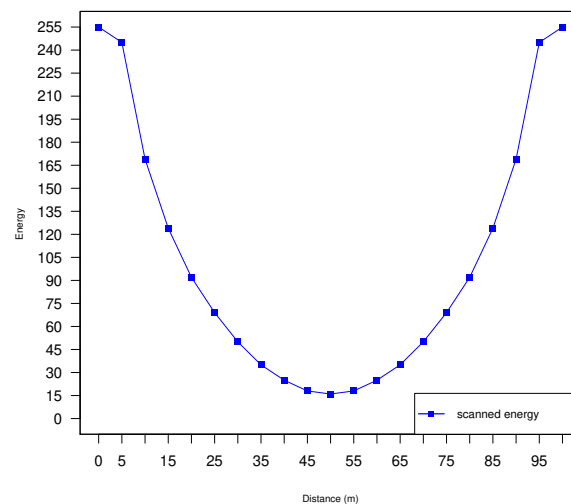


Figure 6. Node energy scan with increasing distance: coordinators set at 0 m and 100 m.

Figure 7 shows the result of a passive scan using a device scanning at different distance intervals. The coordinator generating beacons is at the origin. The current ns-3 LQI implementation for IEEE 802.15.4 uses the signal-to-noise (SNR) as a base to generate its LQI values. As in the previous scenario, the **LogDistancePropagationLossModel** was used. This means that only the distance affects the value of the received signal. Without any additional interference, LQI maintains its highest value (255) until the signal reaches values close to the Rx sensitivity (-106.58 dBm) around 100 m at which point the LQI value rapidly degrades.

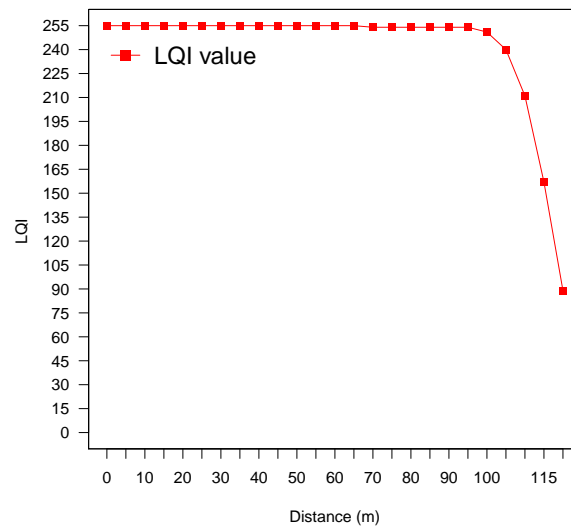


Figure 7. LQI vs distance without interference (2 nodes).

5.2. Association Experiments

In a preliminary experiment, we tested the association of a single device with a CSMA/CA configured with a *random backoff delay* of 0 ($macMinBE = 0$). In our test with these parameters, an association process took at least 0.49805 s (time between the association request and its confirmation) to complete the association with a coordinator. This accounts for the exchange of the six necessary frames and the minimal CSMA/CA operations. In reality, association time is usually higher in a deployment environment because CSMA/CA is generally not set with minimum values, and multiple random backoff delays take place for each transmitted command frame combined with other factors such as frame retransmissions and deference of frames to other CAP caused by contention problems. Regardless of these issues, when the default standard configuration values are used, *macResponseWaitTime* comprises the majority of the association time with a fixed delay of $32 \times aBaseSuperframeDuration$ symbols (0.49152 s) in a 250 kbps O-QPSK PHY (See Figure 4). According to the standard, this value is purposefully high to allow the coordinator to process multiple incoming requests. In the literature, evaluation of the IEEE 802.15.4: association process is often combined with the scanning process [11]. However, this can be an unfair way to evaluate the association performance because a wrongful or disadvantageous configuration of the scanning parameters (e.g., too short or too long scan duration, unnecessary scan of channels) can understandably lead to poor performance of the bootstrap phase. In our evaluations, the association is tested independently from scanning. Devices were configured to be in the communication range of a single coordinator and eliminate any *hidden terminal* problems. Nodes were arranged in a grid pattern and separated from each other by 3 m (Figure 8). Transmissions of *association request commands* were sent by all devices to the coordinator with a specified interval. All experiments used 10 randomized repetitions for each parameter combination. Association experiments used the standard default values in Table 1 unless specified otherwise.

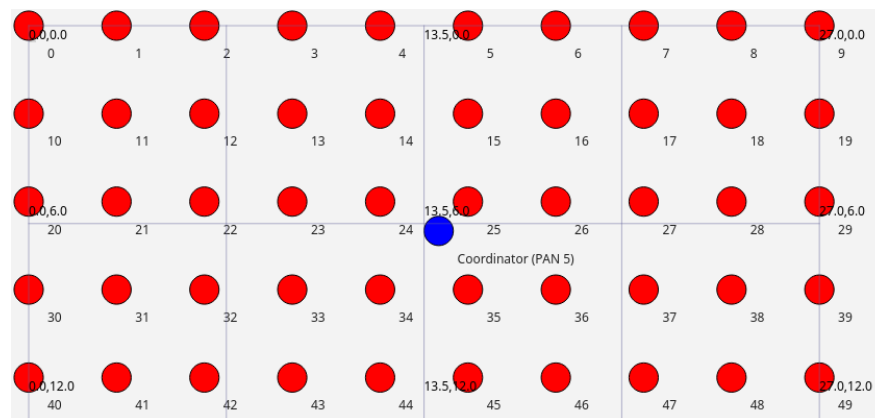


Figure 8. Example of grid topology used in association experiments (50 nodes, 1 coordinator).

Table 1. Experiment parameters.

Parameter	Value
macAckWaitDuration	54 sym (0.000864 s)
SO==BO	3 (0.12288 s)
macMaxCsmaBackoffs	4
macMinBe	3
macResponseWaitTime	30,720 sym (0.49152 s)
assocRespCmdWaitTime	245,760 syms (3.93216 s)
Random Uniform Seed	3
Random Run Number (10 runs)	7~16
macTransactionPersistenceTime	7680 sym (0.12288 s)

Figures 9–11 show the time results of a complete network association using a combination of different association request intervals and several nodes. From Figure 9, we can observe the network association time increase linearly along with the number of nodes and association request intervals. When the association requests interval is reduced, the association time is also drastically reduced. However, short intervals between association requests can lead to longer association times in some situations as observed for intervals 0.01 and 0.02 using a topology of 100 nodes in Figure 10. This is caused by the contention of the channel during the association process. In short, the coordinator does not have enough time to handle all the association requests. Devices are faced with frames getting lost caused by collisions or even the inability to access the channel because of its saturation (Figure 12). Under these circumstances, retransmissions of frames are frequent, and in some cases, a complete failure of the association process occurs. This effect is even more noticeable when many nodes compete for access to the channel within small association intervals as shown in Figure 11. In this figure, the opposite of Figures 9 and 10 is true. As the association request intervals become smaller, the association time tends to increase. This is because the coordinator can no longer timely handle the response to association requests on top of its task involving the reception of these requests and the burden of transmitting beacons. In these situations, requests and responses are delayed, retransmitted, or even fail due to the saturation of the channel. In our experiments, an association failure leads to the restart of the association process after a new association request interval is issued, which impacts the overall network association time. The impact of the association failure is minimal when it occurs in the early stages of the association (during the Tx of the association request command or its ACK); however, this has a significant impact when it occurs in later stages of the association (i.e., after the macResponseWaitTime) as observed in the jumps in association times shown in Figure 11.

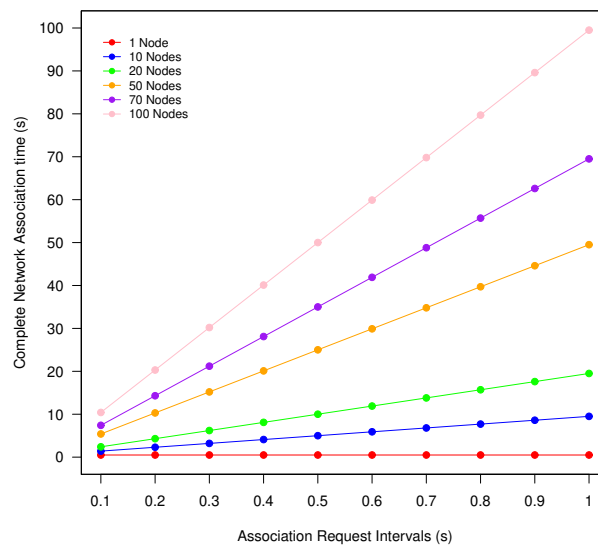


Figure 9. Full network association time (interval 0.1–1 s).

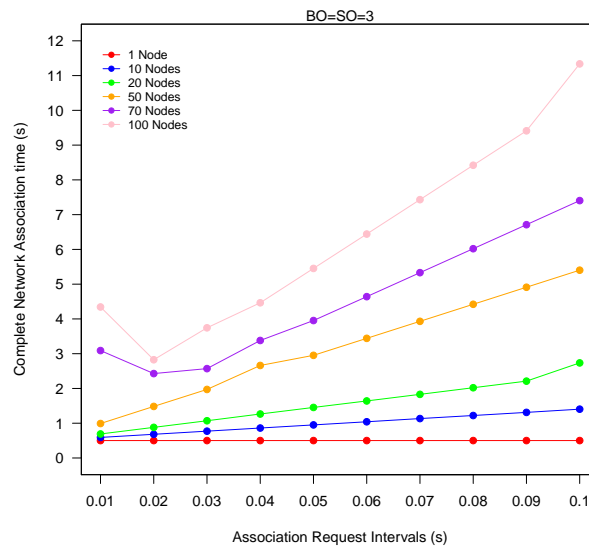


Figure 10. Full network association time (interval 0.01–0.1 s).

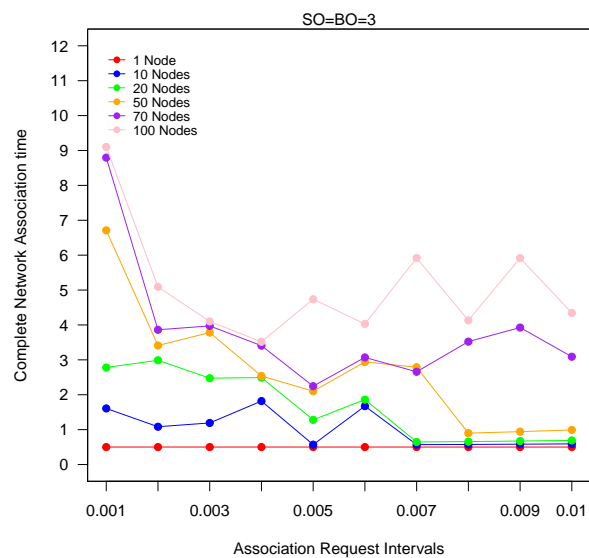


Figure 11. Full network association time (interval 0.001–0.01 s).

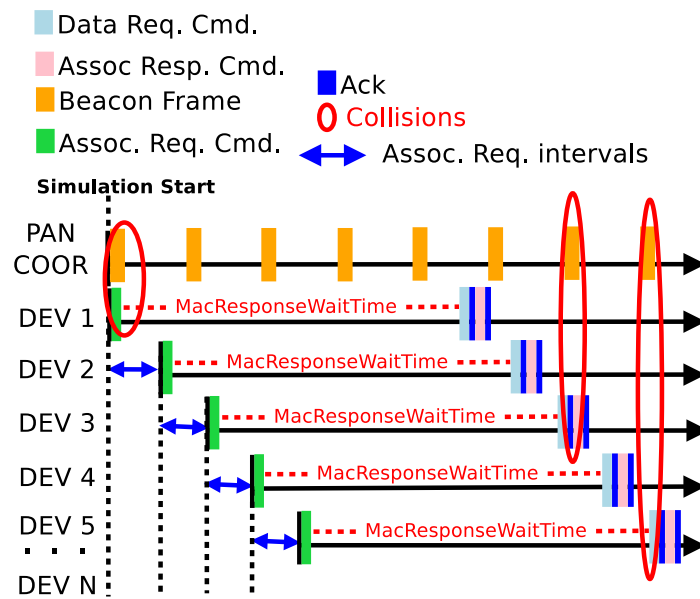


Figure 12. Association command frames overlap.

The frequency at which the coordinator broadcasts beacons has a small impact on the association process. As established in previous sections, variable BO indicates the duration of the maximum beacon interval. The SO variable represents the portion of the interval assigned to the duration of the superframe (beacon period + active period). Therefore, when both variables have the same value, the active period of the superframe is equal to the entire interval between the beacons. In other words, the period in which devices can contend for the channel is equal to the entire period between beacons. The value of the SO variable cannot exceed the value of the variable BO. SO and BO values range between 0 and 14. Table 2 shows the values of these variables in time units and symbols (valid for 250 kbp O-QPSK PHY). Figures 13 and 14 show the results of experiments combining different beacon intervals with different association request intervals on a large network formed by 100 nodes.

Table 2. SO and BO variables time representation (250kbps O-QPSK PHY).

BO & SO	Unit
0	960 sym (1536 μs)
1	1920 sym (3072 μs)
2	3840 sym (6144 μs)
3	7640 sym (122,880 μs)
4	15,360 sym (245,760 μs)
5	30,720 sym (491,520 μs)
6	61,440 sym (983,040 μs)
7	122.88 ksym (1.96608 s)
8	245.76 ksym (3.93216 s)
9	491.52 ksym (7.86432 s)
10	983.04 ksym (15.72864 s)
11	1966.08 ksym (31.45728 s)
12	3932.16 ksym (62.91456 s)
13	7864.32 ksym (125.82912 s)
14	15,728.64 ksym (251.65824 s)

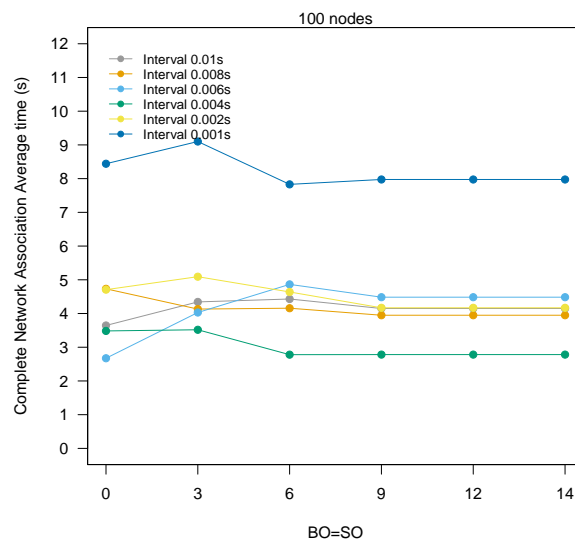


Figure 13. Full network association time with different beacon intervals.

In Figure 13, it is possible to observe a small increase in the association time when the beacon frequency is short (mainly BO = SO = 1–3) in a network of 100 nodes. As the beacon frequency decreases, the impact caused by the beacons also decreases, and the association time becomes regular. This means that while beacon frequency plays a role in the network association time, it is still the interval at which association requests are sent and the number of associating nodes that has the biggest impact on the association time of the network.

Short association request intervals do not guarantee a small association time; in fact, the opposite is typically true. As observed in Figure 13, intervals 0.006 to 0.01 were close in value, interval 0.004 obtained the lowest association time, and finally, association intervals 0.001 s and 0.002 s registered the highest association time and packet overhead (Figure 14). This phenomenon closely correlates with the number of association failures registered (i.e., channel access failures, lost ACK, lost association responses) as shown in Figure 15. Interval 0.004 had the least association failures and therefore the lowest association time and packet overhead. As previously established, association failures cause the reinitialization of the association process and therefore effectively increase both the association time and packet overhead. Ideally, a network of 100 nodes should register only 600 packets for the association process (this accounts for only command packets and ACK). Therefore, when the association request interval is increased as shown in Figure 16, it is possible to obtain values closer to an ideal overhead as shown in the intervals 0.04–0.08 s.

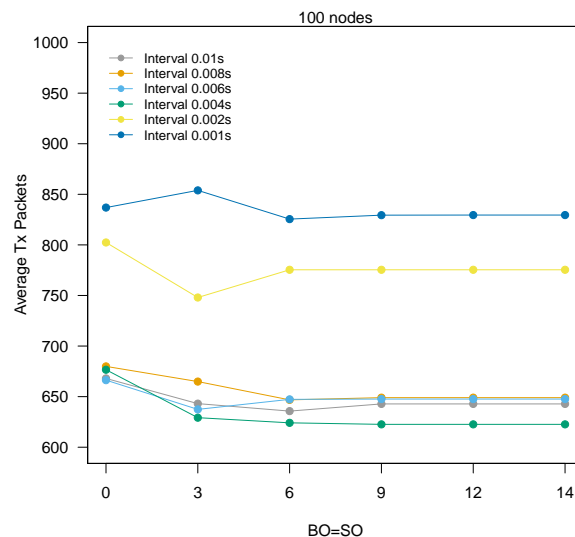


Figure 14. Network Tx packets overhead with different beacon intervals.

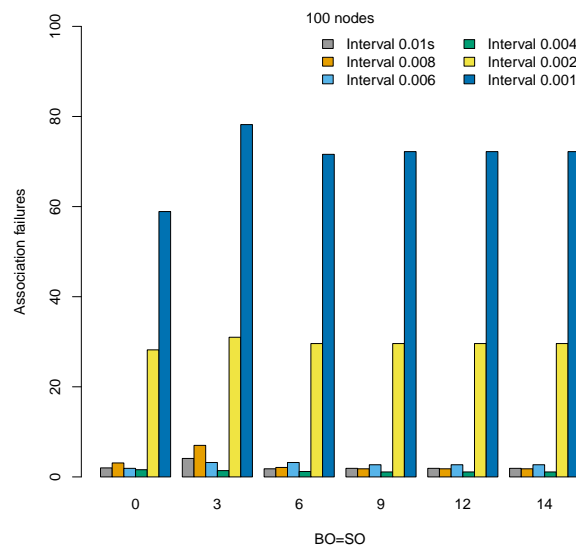


Figure 15. Association request failures (channel access failure, lost ACK, lost response).

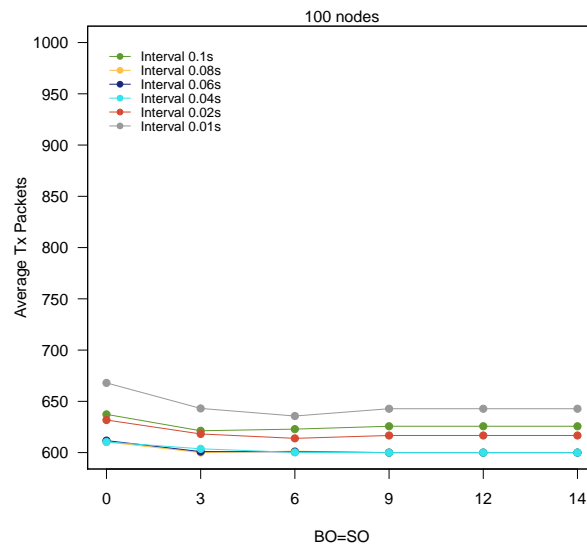


Figure 16. Reduced network Tx packets overhead by increasing the association request intervals.

As described in this work, IEEE 802.15.4 classic association requires the exchange of multiple frames to associate a single device. This process is inherently prone to errors (e.g., beacon storms [19]), and it can be energy inefficient and time-consuming if the packet overhead is high. Not many alternative association proposals exist; however, a few proposals can be found made by authors in [20–22] and in more advanced MACs in [23–25]. The standard addresses directly this problem to some degree in its IEEE 802.15.4e-2012 [26] amendment and is part of the latest revision of the standard [27]. This amendment introduced an alternative way to establish an association (known as *fast association*), which reduces the exchange of frames and avoids the use of indirect transmissions when associating devices; as a consequence, the need for a *pending transaction list* in the association process is also omitted. Regardless of the obvious drawbacks found in the classic association process, the classic association has legacy support in the latest revision of the standard. Furthermore, it is the only available association implementation found in devices still using 2003–2011 revisions of the standard (arguably the most widely found in commercial applications). Finally, we chose the classic association because its inclusion also meant extending ns-3 to support indirect transmissions.

6. Conclusions

In this paper, we implemented the first publicly available implementation and performance evaluation of ns-3's IEEE 802.15.4 bootstrap mechanism. Unlike common belief, scanning and association are independent MAC mechanisms which can be used separately to obtain an optimal association performance. Our evaluations show that the interval at which association requests generate and the number of participating nodes are the defining factors that influence the association mechanism performance in terms of time and packet overhead. The present work can be used to evaluate the optimal values of a PAN deployment using realistic parameters: in essence, the inclusion of the device's initialization phase and its impact on the network for a given configuration.

The work in this paper can be improved by extending it to support the *fast association* mechanism. Improving the flexibility and accuracy of the scanning mechanism can be achieved by adding Rx sensitivity and ED threshold configuration options to ns-3, both of which only use fixed values in the current Lr-wpan module implementation. Making these values as flexible as possible is important to develop new robust ED schemes that are low in complexity and adaptable to noise variance. The proposed implementation was checked for memory leaks, documented, and organized according to the ns-3 coding style. Future works will include memory usage optimizations.

Author Contributions: Conceptualization, A.G.R. and T.N.; methodology, A.G.R.; software, A.G.R.; validation, A.G.R. and T.N.; writing—original draft preparation, A.G.R.; writing—review and editing, A.G.R. and T.N.; supervision, T.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data supporting the reported results can be generated from the source code found at <https://www.nsnam.org/> (accessed on 5 December 2022) on the latest ns-3-dev repository or ns-3 v3.7 and onwards.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Description
WSN	Wireless sensor network
IoT	Internet of Things
PAN	Personal area network
LR-WPAN	Low-rate wireless personal area networks
PHY	Physical layer
MAC	Media access control layer
CSMA/CA	Carrier sense multiple access with collision avoidance
DSME	Deterministic and synchronous multichannel extension
TSCH	Time slotted channel hopping
RPL	Routing protocol for low-power and lossy networks
FFD	Fully functional device
RFD	Reduced functional device
BI	Beacon interval
BO	Beacon order
O-QPSK	Offset quadrature phase-shift keying
SD	Superframe duration
SO	Superframe order
CAP	Contention access period
CFP	Contention-free period
GTS	Guaranteed time slots
ACK	Acknowledgment frame
MCPS	MAC common part sublayer

MLME	MAC sublayer management entity
PD	Physical layer data
PLME	Physical layer management entity
MPDU	MAC protocol data units
FCS	Frame check sequence
BE	Beacon exponent
ED	Energy detection
LQI	Link quality indicator

References

1. *IEEE Std 802.15.4-2003*; IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE: New York, NY, USA, 2003; pp. 1–670. [\[CrossRef\]](#)
2. *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*; IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). IEEE: New York, NY, USA, 2006; pp. 1–320. [\[CrossRef\]](#)
3. *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*; IEEE Standard for Local and Metropolitan area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE: New York, NY, USA, 2011; pp. 1–314. [\[CrossRef\]](#)
4. Connectivity Standards Alliance. Zigbee Specification. 2017. Available online: <https://csa-iot.org/all-solutions/zigbee/> (accessed on 22 July 2022).
5. Thread Group. Thread Network Fundamentals V3.0. Available online: <https://www.threadgroup.org/> (accessed on 22 July 2022).
6. Nsam. Ns-3 Discrete Event Simulator. Available online: <http://www.nsnam.org> (accessed on 15 January 2022).
7. Oikonomou, G.; Duquenois, S.; Elsts, A.; Eriksson, J.; Tanaka, Y.; Tsiftes, N. The Contiki-NG open source operating system for next generation IoT devices. *SoftwareX* **2022**, *18*, 101089. [\[CrossRef\]](#)
8. Elsts, A. TSCH-Sim: Scaling Up Simulations of TSCH and 6TiSCH Networks. *Sensors* **2020**, *20*, 5663. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Nikolaev, S.; Banks, E.; Barnes, P.D.; Jefferson, D.R.; Smith, S. *Pushing the Envelope in Distributed Ns-3 Simulations: One Billion Nodes*; WNS3 '15; Association for Computing Machinery: New York, NY, USA, 2015; pp. 67–74. [\[CrossRef\]](#)
10. Watteyne, T.; Vilajosana, X.; Kerkez, B.; Chraim, F.; Weekly, K.; Wang, Q.; Glaser, S.; Pister, K. OpenWSN: A standards-based low-power wireless development environment. *Trans. Emerg. Telecommun. Technol.* **2012**, *23*, 480–493. [\[CrossRef\]](#)
11. Zheng, J.; Lee, M.J. A comprehensive performance study of IEEE 802.15. 4. *Sens. Netw. Oper.* **2006**, *4*, 218–237.
12. OmNet++ Castalia. Castalia User Manual V3.3. Available online: <https://github.com/boulis/Castalia> (accessed on 9 June 2020).
13. NXP. NXP 802.15.4 User Guide. Available online: <https://www.nxp.com/docs/en/user-guide/JN-UG-3024.pdf> (accessed on 22 June 2016).
14. Baldo, N.; Miozzo, M. Spectrum-aware Channel and PHY layer modeling for ns3. In Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, Pisa, Italy, 20–22 October 2009; pp. 1–8.
15. Reynders, B.; Wang, Q.; Pollin, S. A LoRaWAN Module for Ns-3: Implementation and Evaluation. In Proceedings of the 10th Workshop on Ns-3, Surathkal, India, 13–14 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 61–68. [\[CrossRef\]](#)
16. Bonetto, R.; Bui, N.; Rossi, M.; Zorzi, M. McMAC: A Power Efficient, Short Preamble Multi-Channel Medium Access Control Protocol for Wireless Sensor Networks. In Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques, Desenzano del Garda, Italy, 19–23 March 2012; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Brussels, Belgium, 2012; pp. 383–390.
17. Ramonet, A.G.; Noguchi, T. LR-WPAN: Beacon Enabled Direct Transmissions on Ns-3. In Proceedings of the 2020 the 6th International Conference on Communication and Information Processing, Tokyo, Japan, 27–29 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 115–122. [\[CrossRef\]](#)
18. Chehri, A.; Jeon, G.; Choi, B. Link-Quality Measurement and Reporting in Wireless Sensor Networks. *Sensors* **2013**, *13*, 3066–3076. [\[CrossRef\]](#)
19. Hussain, F.B.; Pyun, J.Y. Coordinator Discovery and Association in Beacon-Enabled IEEE 802.15.4 Network. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 635234. [\[CrossRef\]](#)
20. Constante, L.; Lau, J.; Moraes, R.; Araujo, G.; Montez, C.; Leão, E. Enhanced association mechanism for IEEE 802.15.4 networks. In Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12–15 September 2017; pp. 1–8. [\[CrossRef\]](#)
21. Zhang, F.; Wang, F.; Dai, B.; Li, Y. Performance Evaluation of IEEE 802.15.4 Beacon-Enabled Association Process. In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications—Workshops (Aina Workshops 2008), Okinawa, Japan, 25–28 March 2008; pp. 541–546. [\[CrossRef\]](#)
22. Sheraz, A.; Khan, W.U.; Bangash, J.I.; Ullah, S.I.; Salam, A.; Khan, A.W.; Ahmed, S. Impact of Beacon order and Superframe order on IEEE 802.15. 4 for nodes association in WBAN. *EAI Endorsed Trans. Energy Web* **2018**, *5*, e10. [\[CrossRef\]](#)

23. García Algora, C.M.; Alfonso Reguera, V.; García Fernández, E.M.; Steenhaut, K. Parallel Rendezvous-Based Association for IEEE 802.15.4 TSCH Networks. *IEEE Sens. J.* **2018**, *18*, 9005–9020. [[CrossRef](#)]
24. Wang, L.; Reinhardt, A. A simulative study of network association delays in IEEE 802.15.4e TSCH networks. In Proceedings of the 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Macau, China, 12–15 June 2017; pp. 1–3. [[CrossRef](#)]
25. Mohamadi, M.; Djamaa, B.; Senouci, M.R.; Mellouk, A. FAN: Fast and Active Network Formation in IEEE 802.15.4 TSCH Networks. *J. Netw. Comput. Appl.* **2021**, *183–184*, 103026. [[CrossRef](#)]
26. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*; IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer. IEEE: New York, NY, USA, 2012; pp. 1–225. [[CrossRef](#)]
27. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*; IEEE Standard for Low-Rate Wireless Networks. IEEE: New York, NY, USA, 2020; pp. 1–800. [[CrossRef](#)]